Hopeless Relay Protection for Substation Automation

Kirill Nesterov

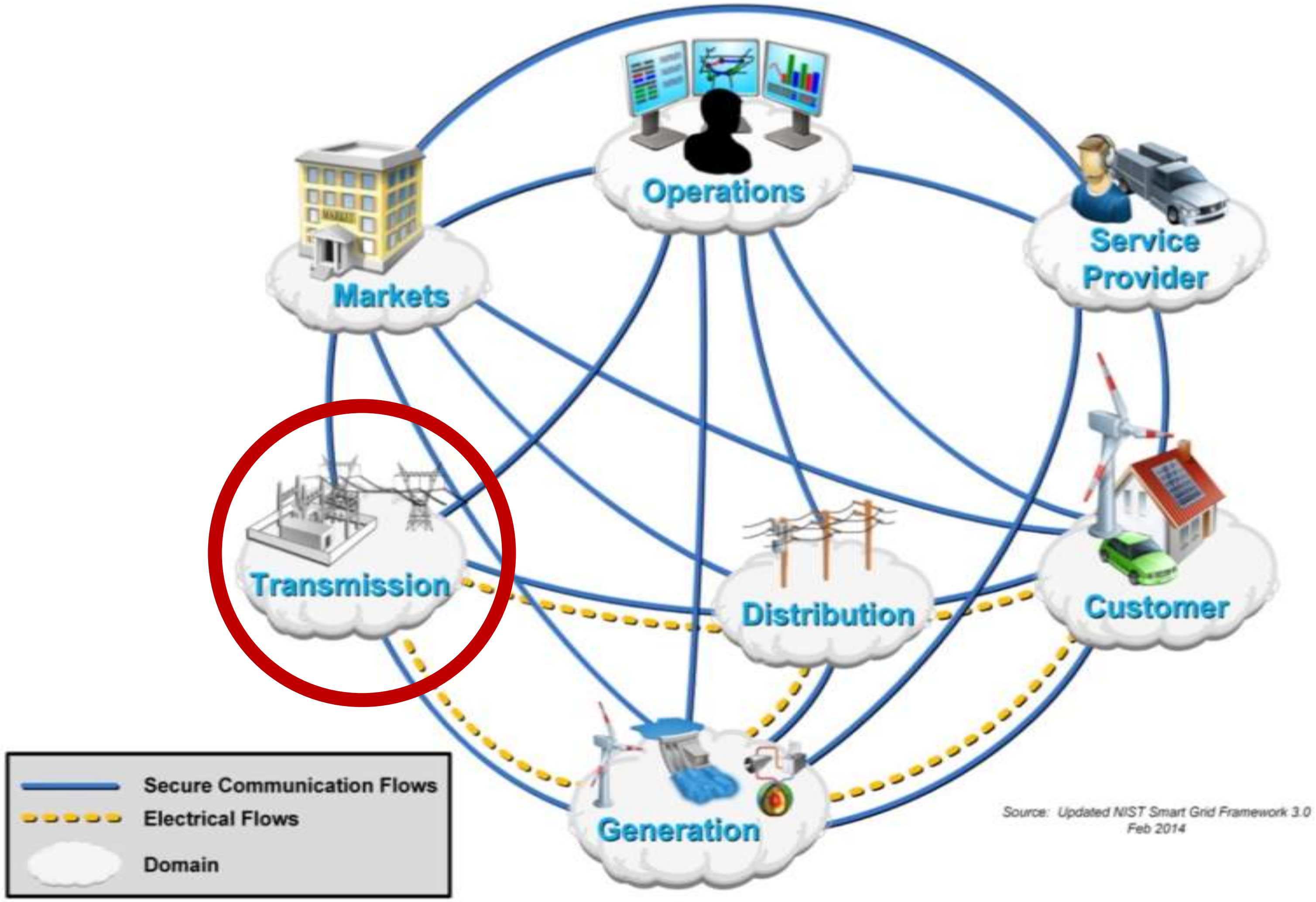
@k_v_nesterov

Alexander Tlyapov

@scadasl

Opinions are my own and not the views of my employer

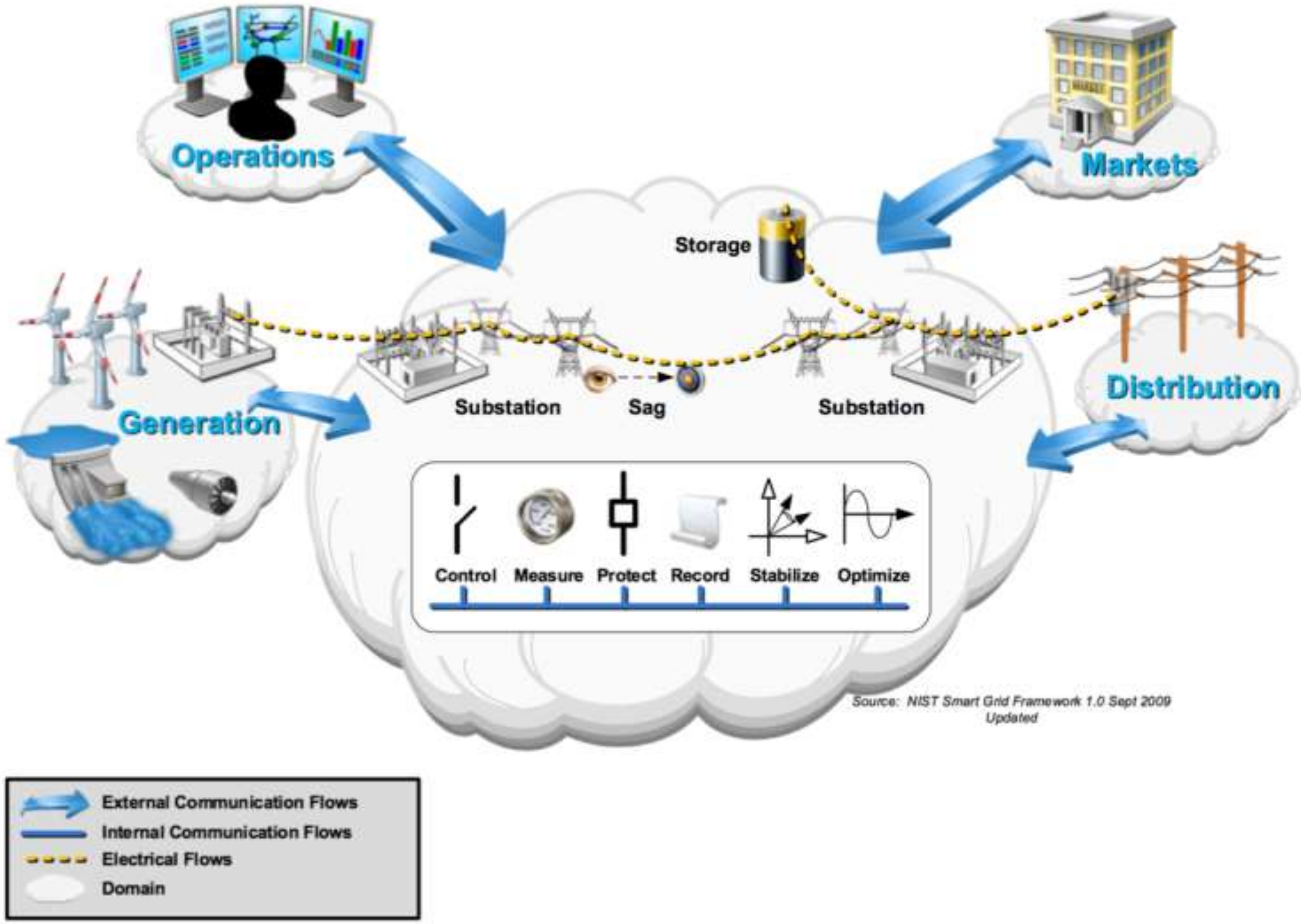
Electric power lifecycle



Source: Updated NIST Smart Grid Framework 3.0
Feb 2014

https://www.nist.gov/sites/default/files/documents/public_affairs/releases/smartgrid_interoperability_final.pdf

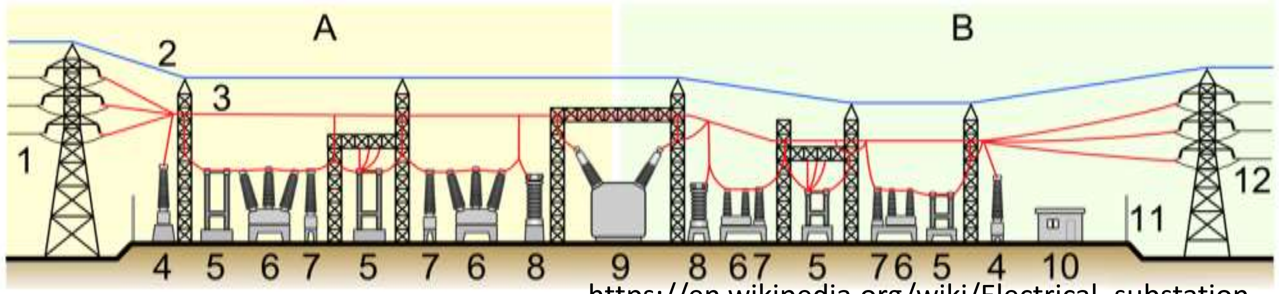
Electric power transmission



https://www.nist.gov/sites/default/files/documents/public_affairs/releases/smartgrid_interoperability_final.pdf

Substation in a nutshell

- Primary system devices
  - Circuit breakers, Disconnect and grounding switches, Power transformers, Instrument transformers, Generators



https://en.wikipedia.org/wiki/Electrical_substation

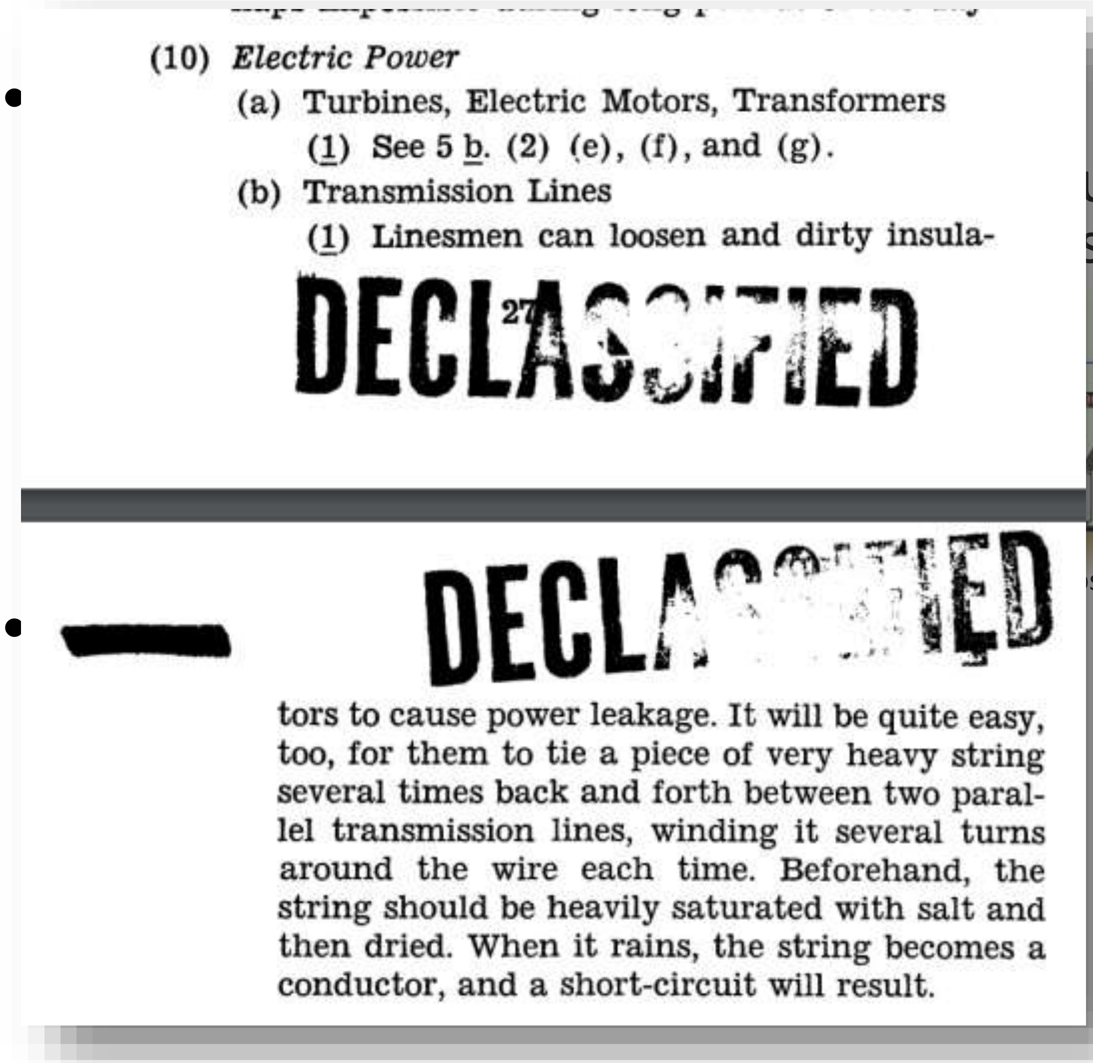
- Secondary system devices
  - Protection, Reclosers, Annunciators, Meters, sensors, Fault recorders, Control switches and interfaces
  - Computers are here!

Substation in a nutshell

- und
  s

- Me

s://en.w



(10) *Electric Power*

(a) Turbines, Electric Motors, Transformers

(1) See 5 b. (2) (e), (f), and (g).

(b) Transmission Lines

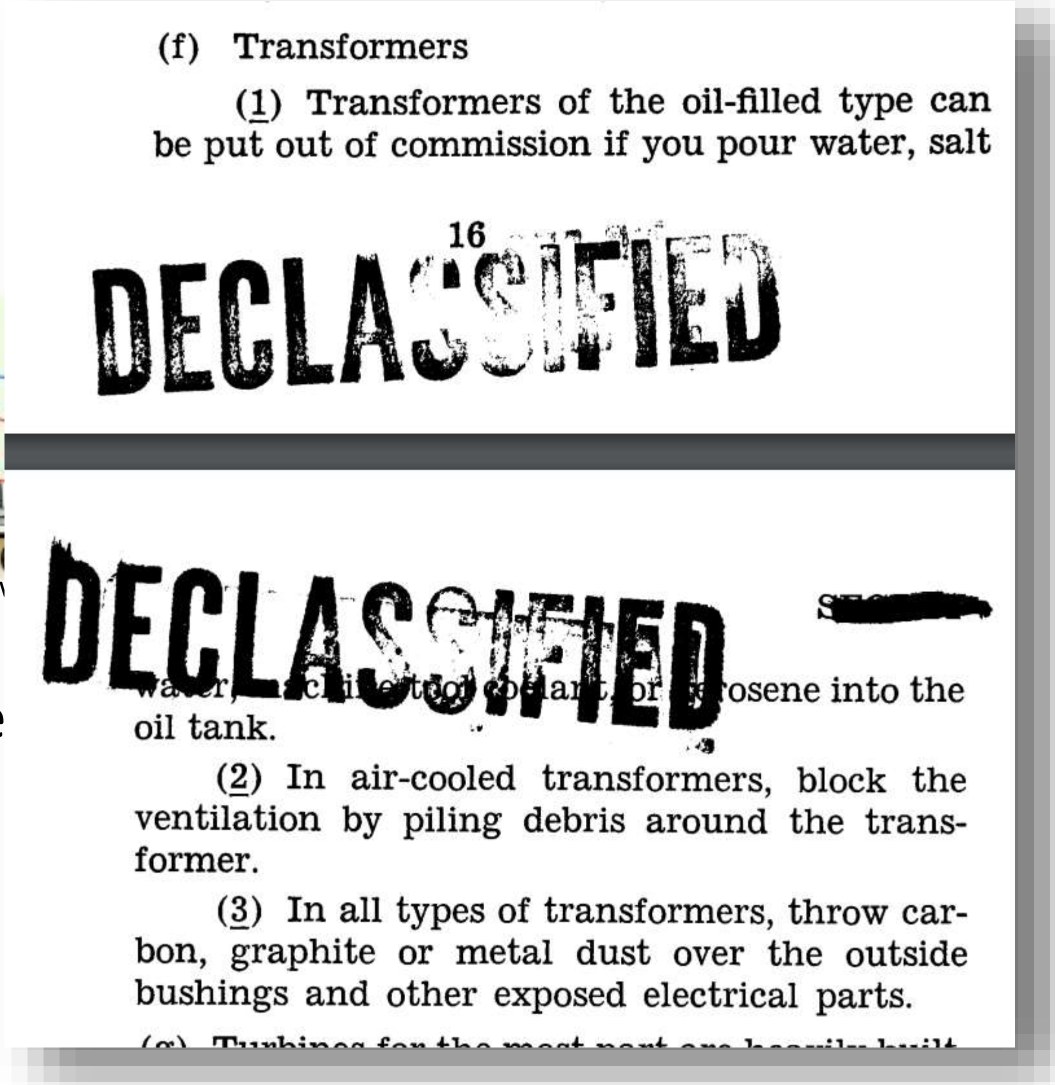
(1) Linesmen can loosen and dirty insula-

DECLASSIFIED

[redacted]

DECLASSIFIED

tors to cause power leakage. It will be quite easy, too, for them to tie a piece of very heavy string several times back and forth between two parallel transmission lines, winding it several turns around the wire each time. Beforehand, the string should be heavily saturated with salt and then dried. When it rains, the string becomes a conductor, and a short-circuit will result.



(f) Transformers

(1) Transformers of the oil-filled type can be put out of commission if you pour water, salt

DECLASSIFIED

DECLASSIFIED

[redacted]

water, machine tool coolant, or kerosene into the oil tank.

(2) In air-cooled transformers, block the ventilation by piling debris around the transformer.

(3) In all types of transformers, throw carbon, graphite or metal dust over the outside bushings and other exposed electrical parts.

(g) Turbines for the most part are heavily built

Small demo

Substation in a nutshell

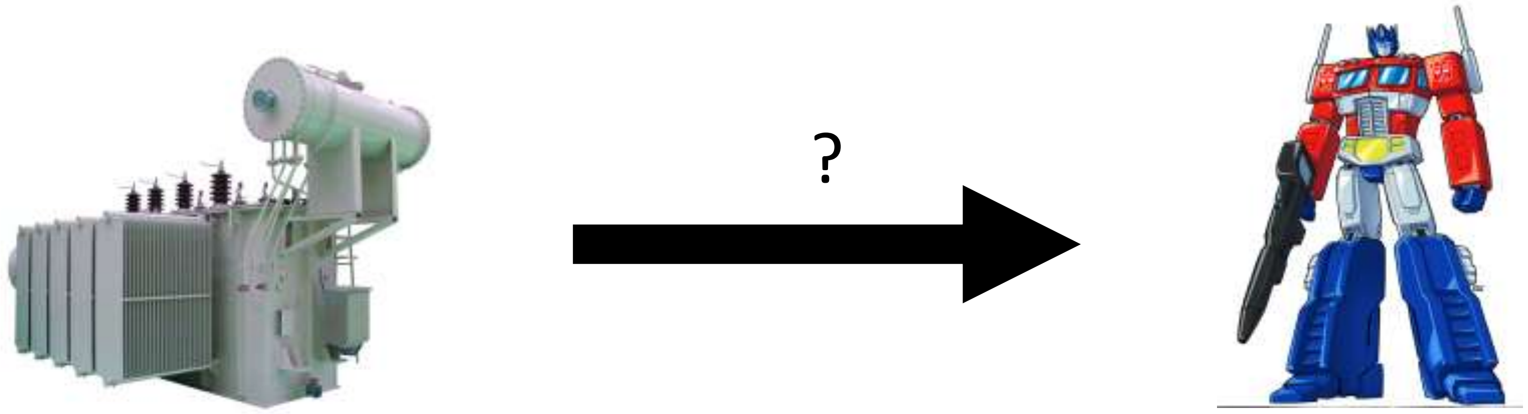
- Everything is in IEC 61850
    - Set of protocols (GOOSE, MMS, SV, etc.)
    - Dafaq Substation Configuration Language (SCL)
- Digital Protective Relay (also IEDs)
- Network bacchanalia
    - Interconnections with substation
        - System operator, Billing, Transmission support
    - Ethernet, Power Line Communications (PLC)

Security of substations

- IEC 61850
  - tldr; **No security**
  - Exploiting the GOOSE Protocol: A Practical Attack on Cyber-infrastructure by Juan Hoyos, Mark Dehus, Timthy X Brown
  - Poisoned GOOSE: Exploiting the GOOSE Protocol http://crpit.com/confpapers/CRPITV149Kush.pdf
- IEC 62351
  - tldr; **use No security via SSH tunnel**
  - Set of words to encapsulate everything from IEC 61850 in encryption
  - Haha, you know, distribution owners update and vendors provide updates

Antiviolence reminder: transformers and geoshmalitics

- Santa Barbabararaba is in another universe
- We are not electrical engineers and that is not the point of the talk
- Yes, we heard that transformer is not like Optimus Prime
    - They just didn’t saw them transforming
    - While colors don’t match, Eleron gas source is planet Cybertron!
- If you want bash us for electrical misanything - just call your therapist



Generic Relay Terminal Internals

- PowerPC (MPC860)
- RTOS
- Protocols
  - IEC61850 (MMSLite)
  - Proprietary protocol for updates
  - Optional Web
- Poor debug facilities
- Today’s menu
  - En salada la Switzerland, Germany, France, USA

SIPROTEC 7

ЯЧХЗЦФ
SIPROTEC
7SJ63
RUN
ERROR
MENU
LED
ESC
ENTER
Event Log
Metering
Trip Log
F1
F2
F3
F4
Local
Remote
Test
Normal

Target device – SIPROTEC 7SJ64x

The software is divided into two main parts:

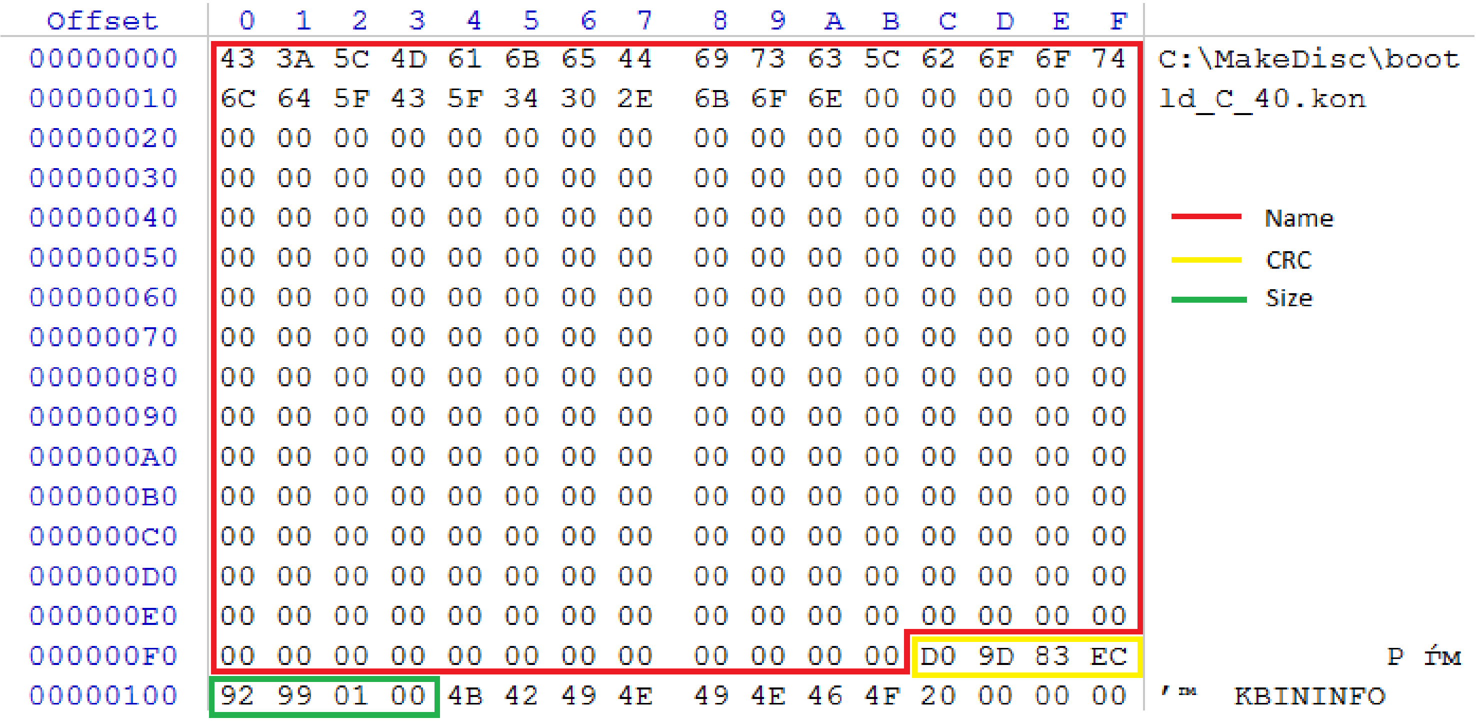
- Common firmware (bootloader, RTOS pSOS+ code, …)
- Modules that implement additional protocols (IEC61850, DNP3, Modbus, …)

The firmware is available as a file with the extension ".PCK" included with the application for the installation - FIRMWAREUPDATE.EXE
PCK File is a container with .KON files, xml with update options and soon

PCK file format

Contains records with file description

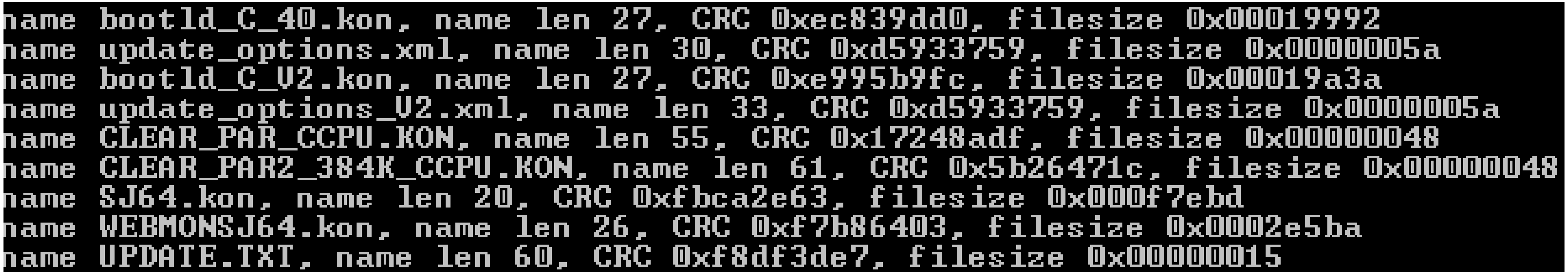
```
struct PCK_file_record
{
   char          Name[252];
   DWORD               CRC;
   DWORD              Size;
}
```



```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  43 3A 5C 4D 61 6B 65 44  69 73 63 5C 62 6F 6F 74  C:\MakeDisc\boot
00000010  6C 64 5F 43 5F 34 30 2E  6B 6F 6E 00 00 00 00 00  ld_C_40.kon
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000000F0  00 00 00 00 00 00 00 00  00 00 00 00 D0 9D 83 EC              Р ѓм
00000100  92 99 01 00 4B 42 49 4E  49 4E 46 4F 20 00 00 00  '™  KBININFO
```

PCK file format

Files in 7SJ64X_04.93.01.PCK



```
name bootld_C_40.kon, name len 27, CRC 0xec839dd0, filesize 0x00019992
name update_options.xml, name len 30, CRC 0xd5933759, filesize 0x0000005a
name bootld_C_V2.kon, name len 27, CRC 0xe995b9fc, filesize 0x00019a3a
name update_options_V2.xml, name len 33, CRC 0xd5933759, filesize 0x0000005a
name CLEAR_PAR_CCPU.KON, name len 55, CRC 0x17248adf, filesize 0x00000048
name CLEAR_PAR2_384K_CCPU.KON, name len 61, CRC 0x5b26471c, filesize 0x00000048
name SJ64.kon, name len 20, CRC 0xfbca2e63, filesize 0x000f7ebd
name WEBMONSJ64.kon, name len 26, CRC 0xf7b86403, filesize 0x0002e5ba
name UPDATE.TXT, name len 60, CRC 0xf8df3de7, filesize 0x00000015
```

Code stored in KON files. One PCK file may contain KON files for different CPU. In this example we have bootloader variants for CCPU and 384K.

KON file format

KON file is set of tagged records with different types. Structure of the record header:

```
struct KON_file_header
{
  char                          Signature[4];
  KON_section_header            Sections[];
}
```

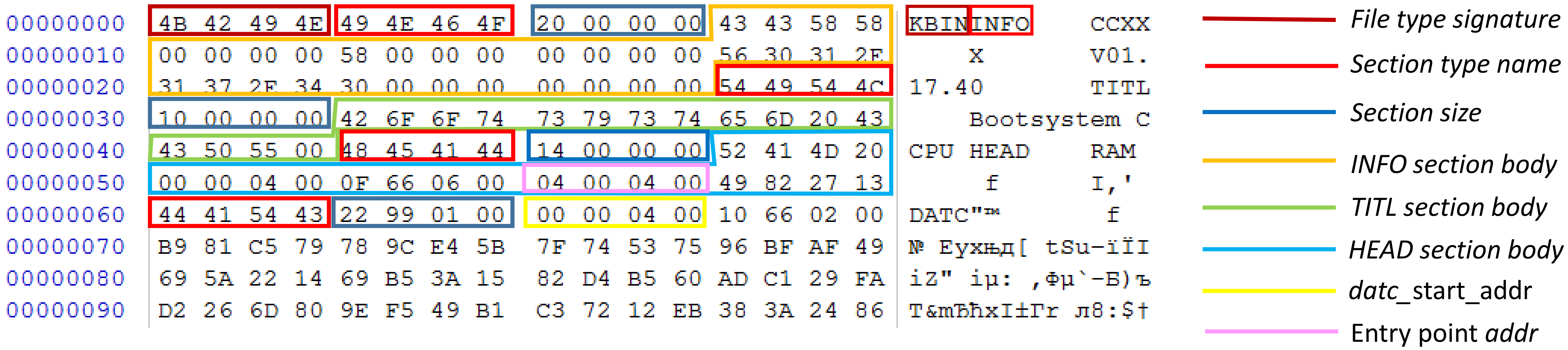
```
struct KON_section_header
{
  char              SectionTypeName[4];
  DWORD             size;
}
```

In the present case we had the following types of records:

- “HEAD” (char code_type*[4]*; DWORD minaddr; DWORD maxaddr; DWORD entry_point ; DWORD xorcks)
- “INFO” (char unit[8]; char device[8]; char version[15]; BYTE number)
- "TITL“ (char title[])
- "DATC“ (DWORD datc_start_addr; DWORD datc_size; DWORD crc)
- "DATA“ (DWORD start_addr; DWORD datca_size; DWORD crc)
- "ENDE“
- **https://github.com/rigmar/Recon2017/tree/master/SIPROTEC**

KON file format

File header of “bootld_C_V2.kon” from firmware:

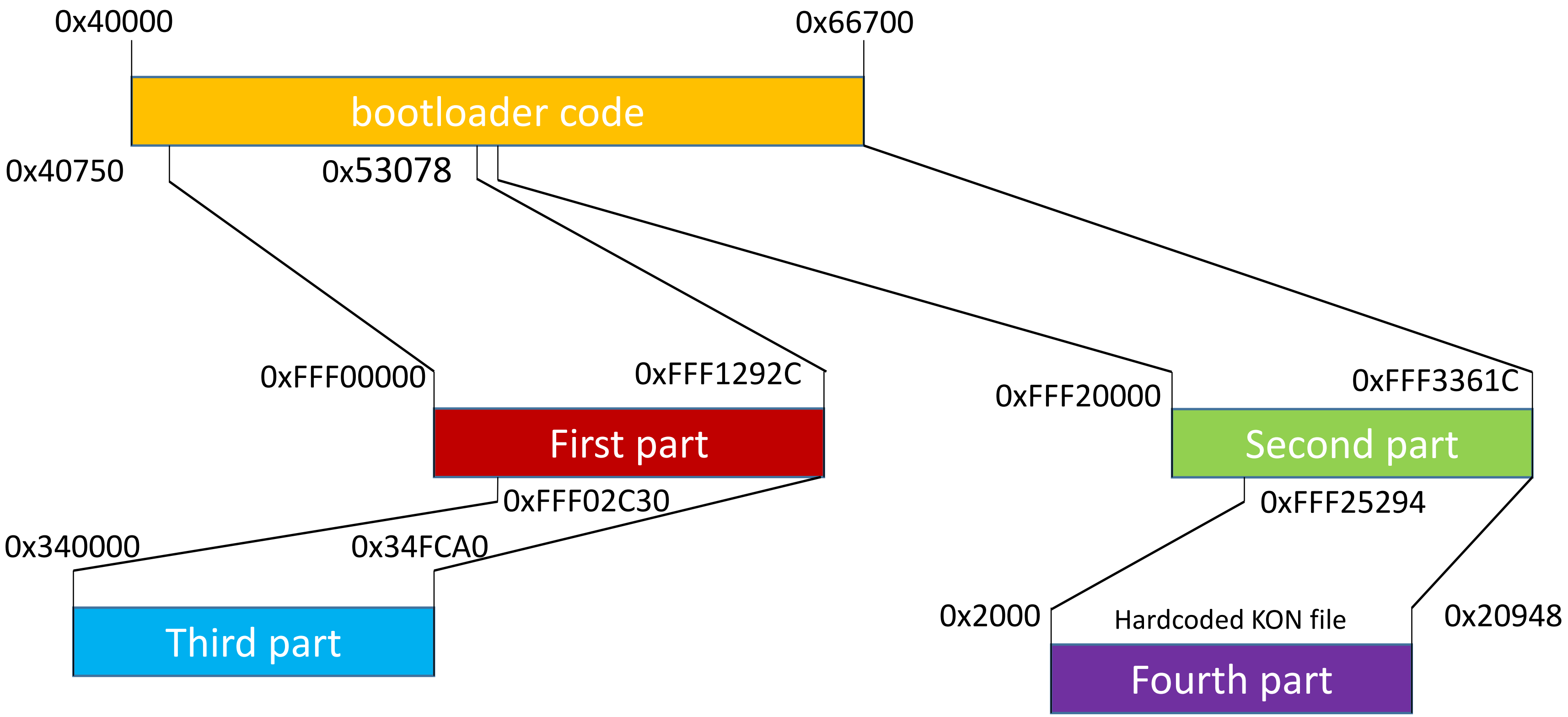


```
00000000  4B 42 49 4E 49 4E 46 4F  20 00 00 00 43 43 58 58  KBININFO    CCXX
00000010  00 00 00 00 58 00 00 00  00 00 00 00 56 30 31 2E      X       V01.
00000020  31 37 2E 34 30 00 00 00  00 00 00 00 54 49 54 4C  17.40       TITL
00000030  10 00 00 00 42 6F 6F 74  73 79 73 74 65 6D 20 43      Bootsystem C
00000040  43 50 55 00 48 45 41 44  14 00 00 00 52 41 4D 20  CPU HEAD    RAM
00000050  00 00 04 00 0F 66 06 00  04 00 04 00 49 82 27 13       f      I,'
00000060  44 41 54 43 22 99 01 00  00 00 04 00 10 66 02 00  DATC"™      f
00000070  B9 81 C5 79 78 9C E4 5B  7F 74 53 75 96 BF AF 49  № Eyxњд[ tSu–їЇI
00000080  69 5A 22 14 69 B5 3A 15  82 D4 B5 60 AD C1 29 FA  iZ" iµ: ‚Фµ`-Б)ъ
00000090  D2 26 6D 80 9E F5 49 B1  C3 72 12 EB 38 3A 24 86  T&mЂћxI±Гr л8:$†
```

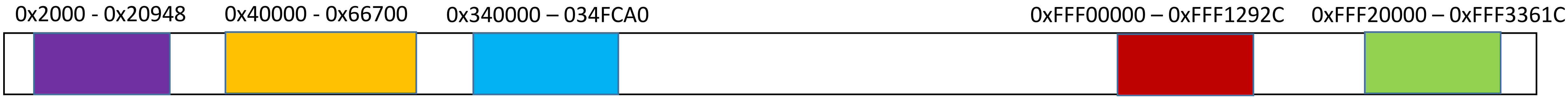
So, we know bootloader start address (0x40000) and entry point (0x40004). Trying to load in IDA PRO and see some problem:

- Part of code has different than 0x40000 base address
- ANOTHER part of code has base address that is different from the previous two

Nested Doll - Matreshka from Germany



Finally bootloader code map memory view



Architecture and OS of device

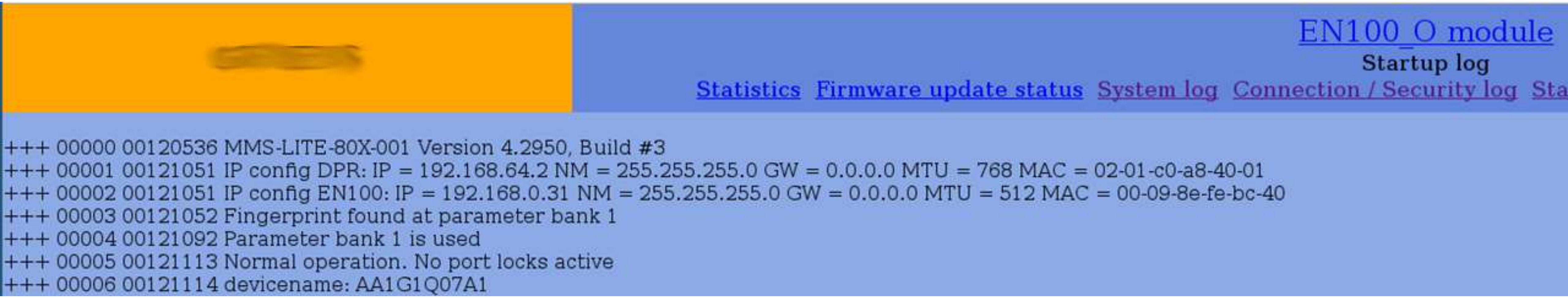
- pSOS+/PPC V2.0.7
- Upgrade with Ethernet module EN100
  - Same CPU
  - Same OS
  - TCP/IP communication
  - Port forwarding

Services

- HTTP (80/tcp)
  - Diagnostics and bonus features!
- DIGSI (5000x/tcp)
  - Proprietary engineering protocol
- Java Applet Remote Managing protocol (56797/udp)
  - Diagnostics
- IEC61850 MMS (102/tcp) and GOOSE
  - Industrial process

Web Server

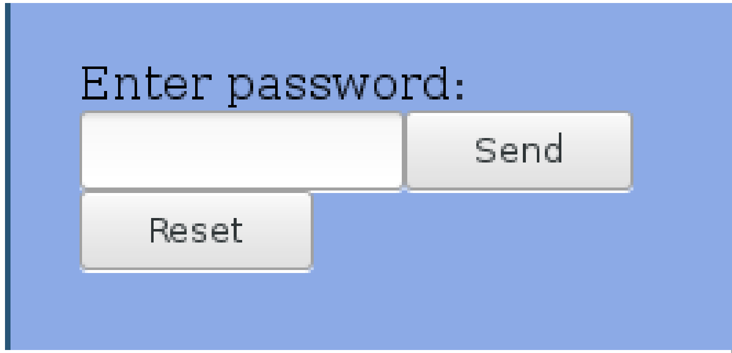
- It’s always a good idea to wright your own

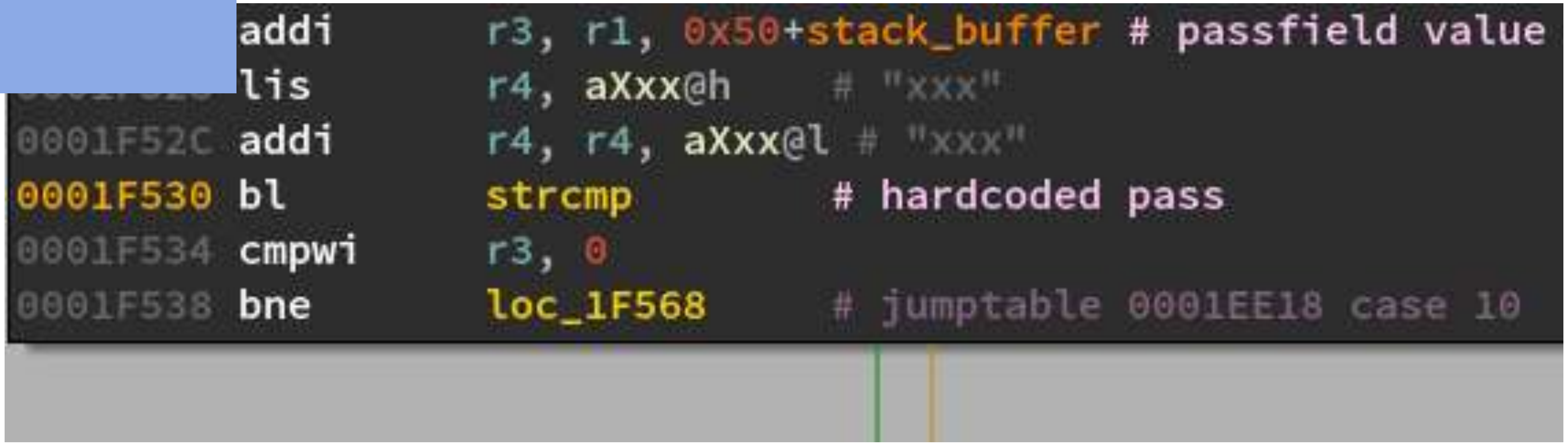


```
+++ 00000 00120536 MMS-LITE-80X-001 Version 4.2950, Build #3
+++ 00001 00121051 IP config DPR: IP = 192.168.64.2 NM = 255.255.255.0 GW = 0.0.0.0 MTU = 768 MAC = 02-01-c0-a8-40-01
+++ 00002 00121051 IP config EN100: IP = 192.168.0.31 NM = 255.255.255.0 GW = 0.0.0.0 MTU = 512 MAC = 00-09-8e-fe-bc-40
+++ 00003 00121052 Fingerprint found at parameter bank 1
+++ 00004 00121092 Parameter bank 1 is used
+++ 00005 00121113 Normal operation. No port locks active
+++ 00006 00121114 devicename: AA1G1Q07A1
```

How to secure your web?

- Password of course!
- CVE-2016-7112

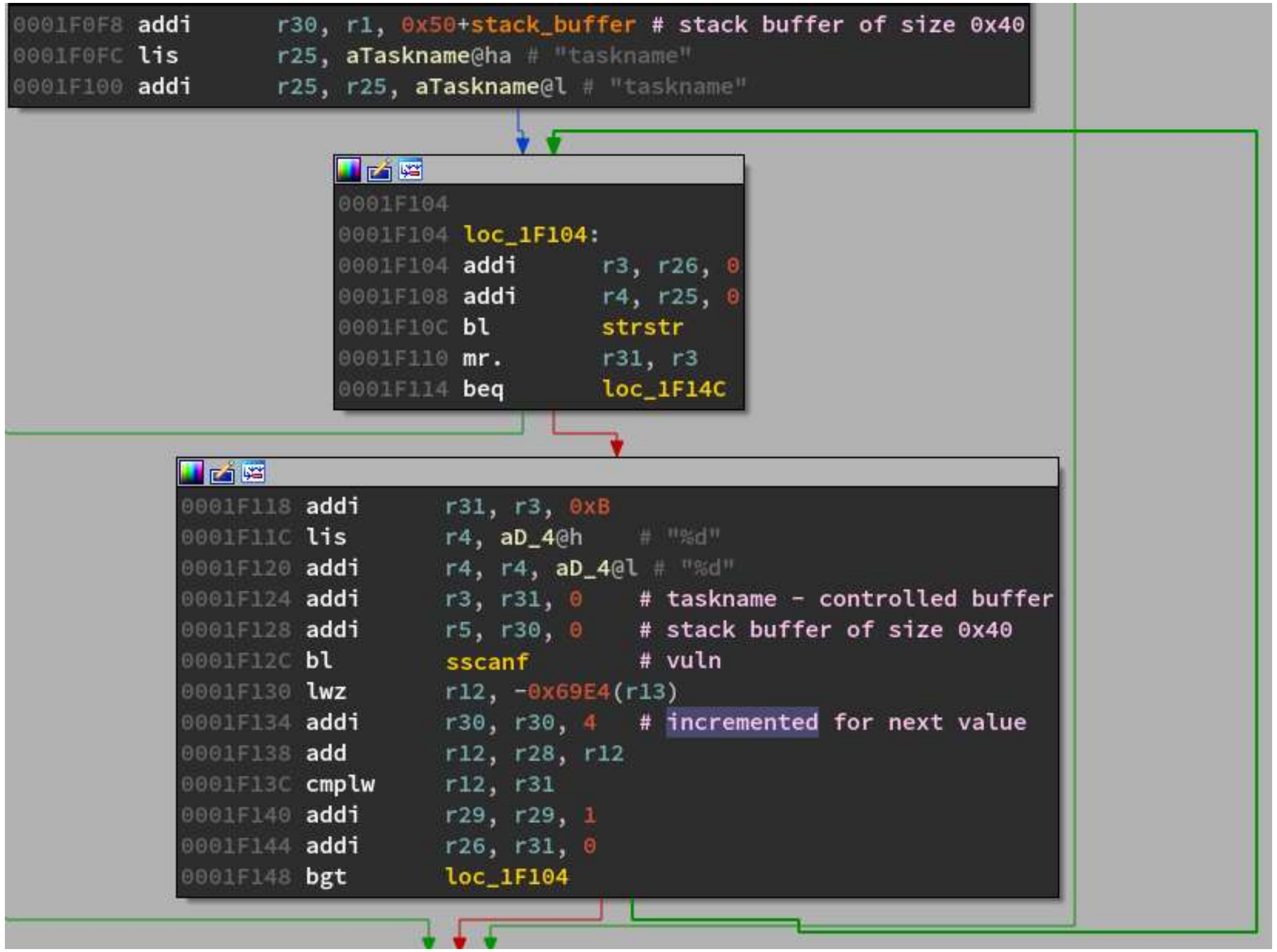




```
          addi      r3, r1, 0x50+stack_buffer # passfield value
          lis       r4, aXxx@h     # "xxx"
0001F52C  addi      r4, r4, aXxx@l # "xxx"
0001F530  bl        strcmp         # hardcoded pass
0001F534  cmpwi     r3, 0
0001F538  bne       loc_1F568      # jumptable 0001EE18 case 10
```

Is your web secure?

- strstr “taskname”
- scanf “%d” into stack
- No canary
- What could go wrong?



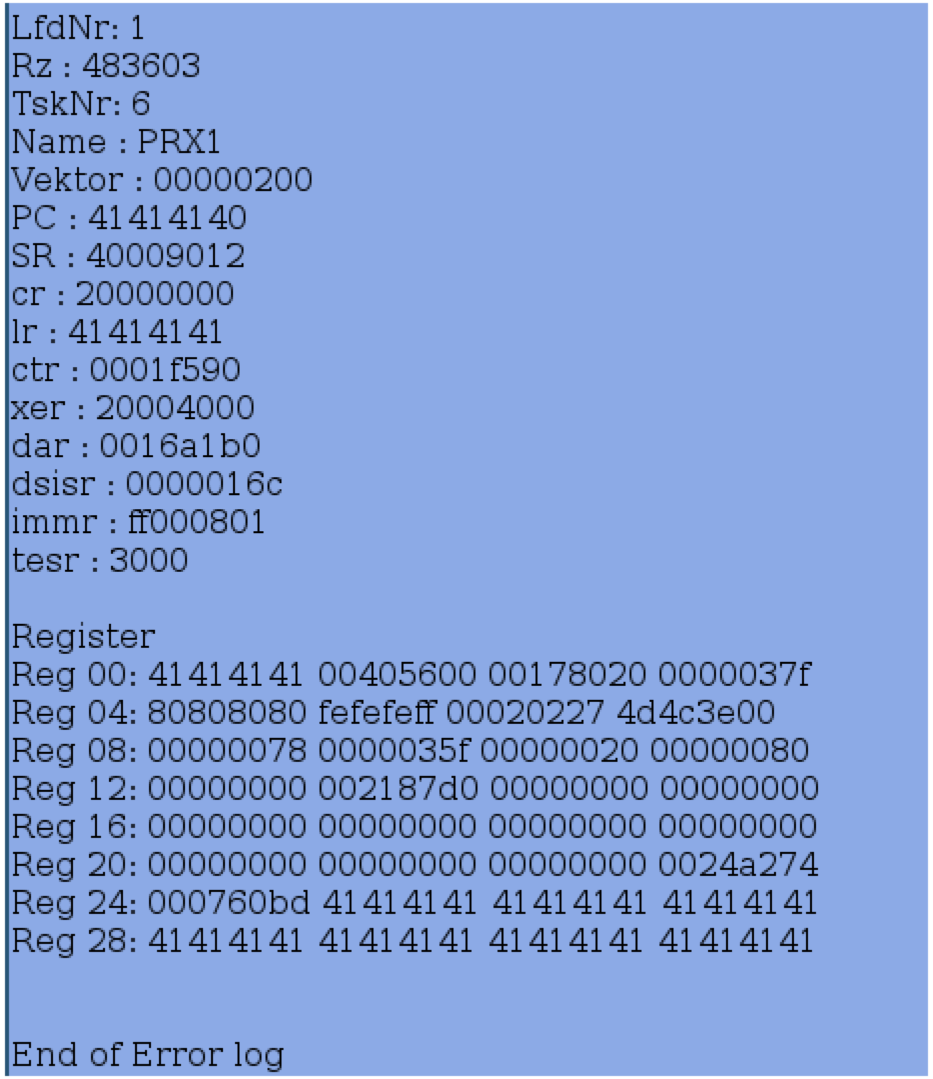
```
0001F0F8 addi      r30, r1, 0x50+stack_buffer # stack buffer of size 0x40
0001F0FC lis       r25, aTaskname@ha # "taskname"
0001F100 addi      r25, r25, aTaskname@l # "taskname"
```

```
0001F104
0001F104 loc_1F104:
0001F104 addi      r3, r26, 0
0001F108 addi      r4, r25, 0
0001F10C bl        strstr
0001F110 mr.       r31, r3
0001F114 beq       loc_1F14C
```

```
0001F118 addi      r31, r3, 0xB
0001F11C lis       r4, aD_4@h    # "%d"
0001F120 addi      r4, r4, aD_4@l # "%d"
0001F124 addi      r3, r31, 0    # taskname - controlled buffer
0001F128 addi      r5, r30, 0    # stack buffer of size 0x40
0001F12C bl        sscanf        # vuln
0001F130 lwz       r12, -0x69E4(r13)
0001F134 addi      r30, r30, 4   # incremented for next value
0001F138 add       r12, r28, r12
0001F13C cmplw     r12, r31
0001F140 addi      r29, r29, 1
0001F144 addi      r26, r31, 0
0001F148 bgt       loc_1F104
```

Complicated auth

- /fehler – error log URL ->
- Very convenient
- Looks promising/pwnable
- PC = 0x41414140
- Network buffers looks RWX



```
LfdNr: 1
Rz : 483603
TskNr: 6
Name : PRX1
Vektor : 00000200
PC : 41414140
SR : 40009012
cr : 20000000
lr : 41414141
ctr : 0001f590
xer : 20004000
dar : 0016a1b0
dsisr : 0000016c
immr : ff000801
tesr : 3000

Register
Reg 00: 41414141 00405600 00178020 0000037f
Reg 04: 80808080 fefefeff 00020227 4d4c3e00
Reg 08: 00000078 0000035f 00000020 00000080
Reg 12: 00000000 002187d0 00000000 00000000
Reg 16: 00000000 00000000 00000000 00000000
Reg 20: 00000000 00000000 00000000 0024a274
Reg 24: 000760bd 41414141 41414141 41414141
Reg 28: 41414141 41414141 41414141 41414141


End of Error log
```

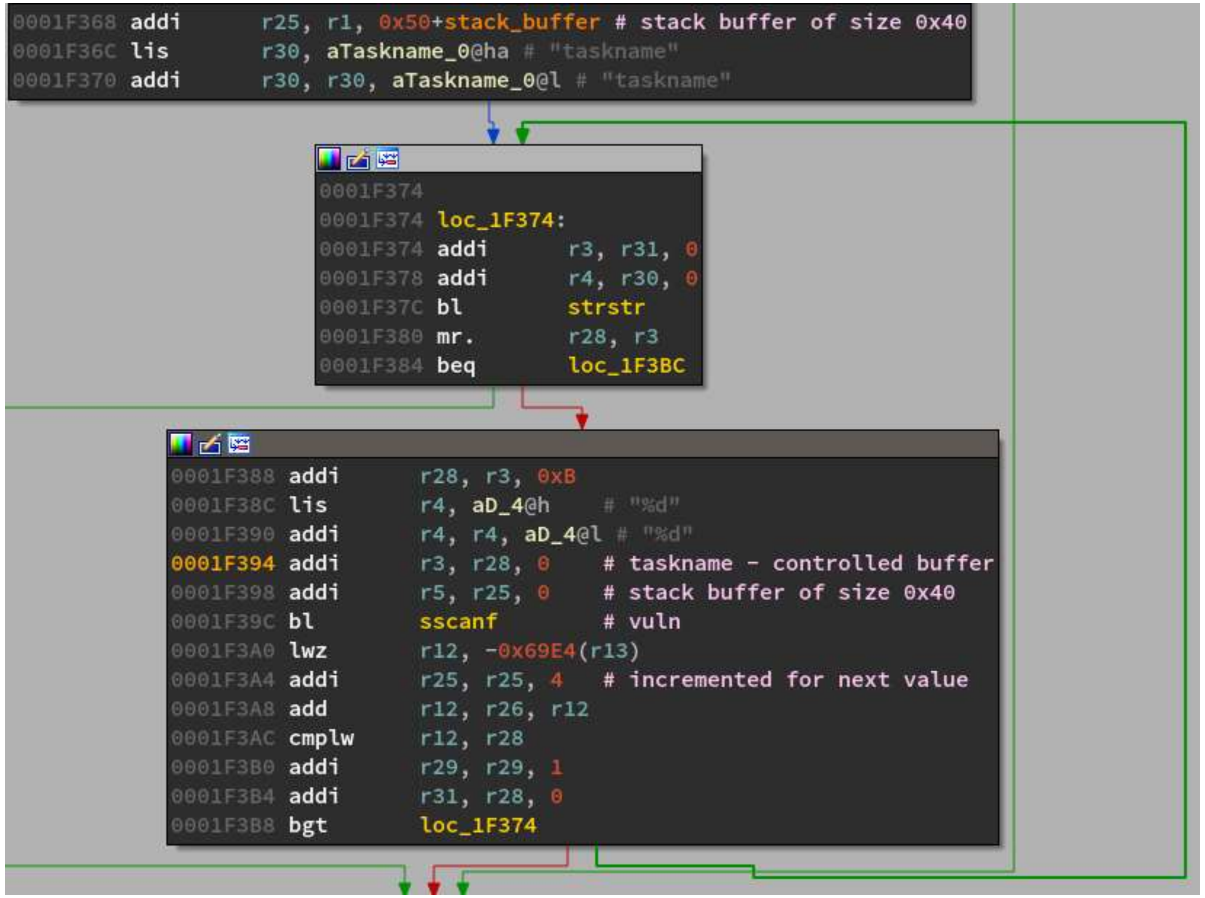
Complicated CVE

- CVE-2016-7113
- CVSS v3.0 Base Score 5.3

DEMO

sscanf problem

- Nearly every call is vulnerable

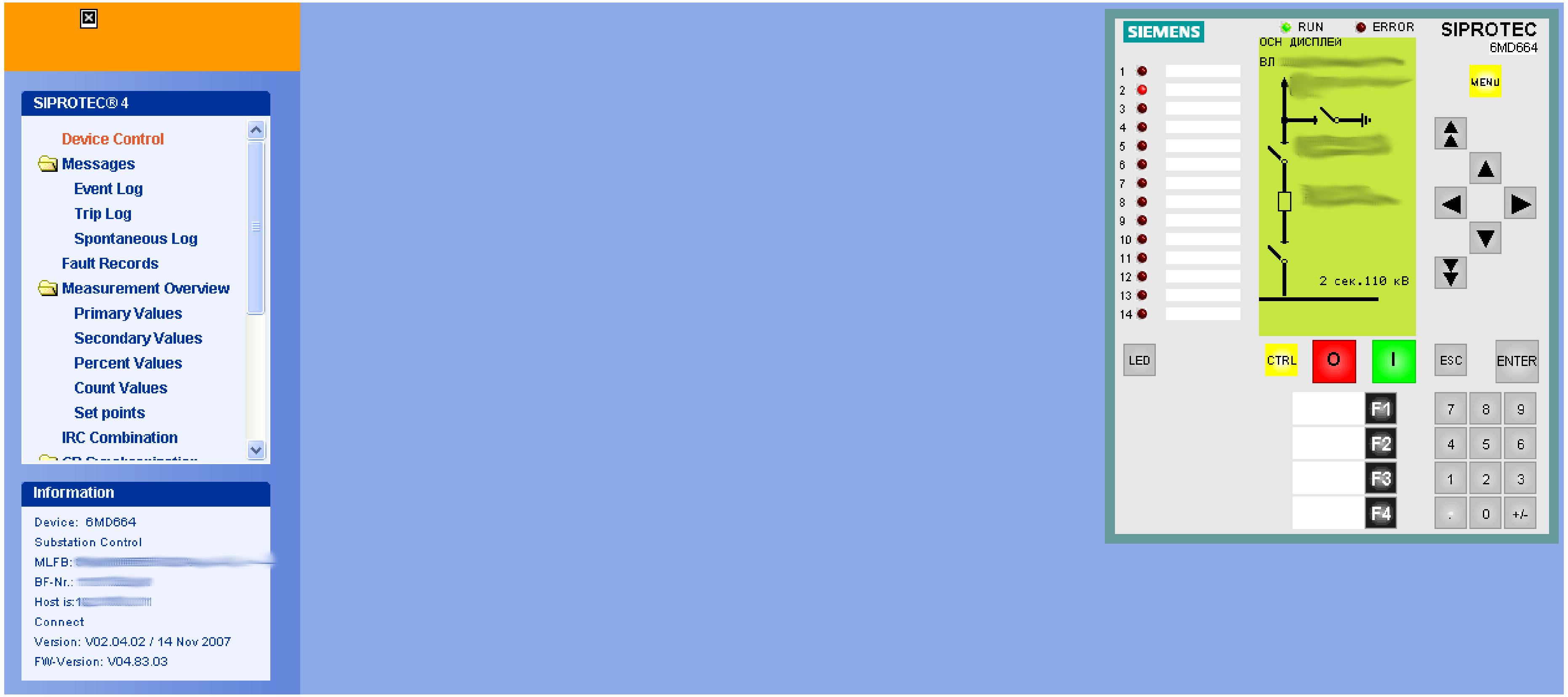


```
0001F368 addi    r25, r1, 0x50+stack_buffer # stack buffer of size 0x40
0001F36C lis     r30, aTaskname_0@ha # "taskname"
0001F370 addi    r30, r30, aTaskname_0@l # "taskname"
```

```
0001F374
0001F374 loc_1F374:
0001F374 addi    r3, r31, 0
0001F378 addi    r4, r30, 0
0001F37C bl      strstr
0001F380 mr.     r28, r3
0001F384 beq     loc_1F3BC
```

```
0001F388 addi    r28, r3, 0xB
0001F38C lis     r4, aD_4@h     # "%d"
0001F390 addi    r4, r4, aD_4@l # "%d"
0001F394 addi    r3, r28, 0     # taskname - controlled buffer
0001F398 addi    r5, r25, 0     # stack buffer of size 0x40
0001F39C bl      sscanf         # vuln
0001F3A0 lwz     r12, -0x69E4(r13)
0001F3A4 addi    r25, r25, 4    # incremented for next value
0001F3A8 add     r12, r26, r12
0001F3AC cmplw   r12, r28
0001F3B0 addi    r29, r29, 1
0001F3B4 addi    r31, r28, 0
0001F3B8 bgt     loc_1F374
```

Java applet



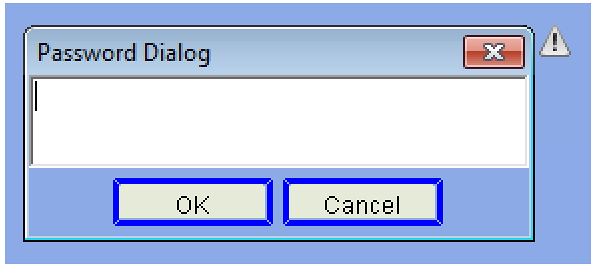
Information

Device: 6MD664
Substation Control
MLFB: [redacted]
BF-Nr.: [redacted]
Host is:1[redacted]
Connect
Version: V02.04.02 / 14 Nov 2007
FW-Version: V04.83.03

High-voltage Bay Controller Unit

Java applet

- CPU service
- Some proprietary 56797/udp protocol
- Some diagnostics
- Some password check on user-side
  - But it's not hardcoded ☹
  - It's confirmation code 311299
- Some read out of bounds => crash



Crash >= RCE

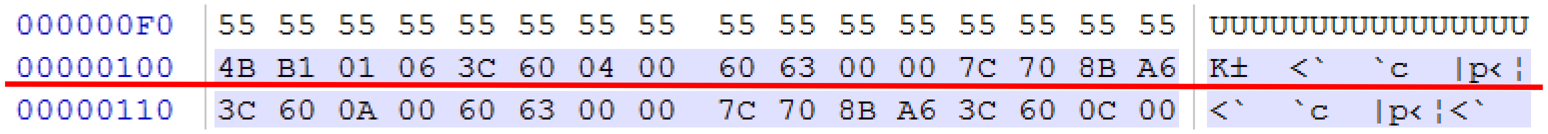
- Defective mode
  - Can be fixed only with manual reboot
- No protection
  - Terminal runs in “Monitor mode”
  - Tested with RETOM device
- True for core CPU bugs

Relay protection and automation testing system

D/F60 Feeder Protection System

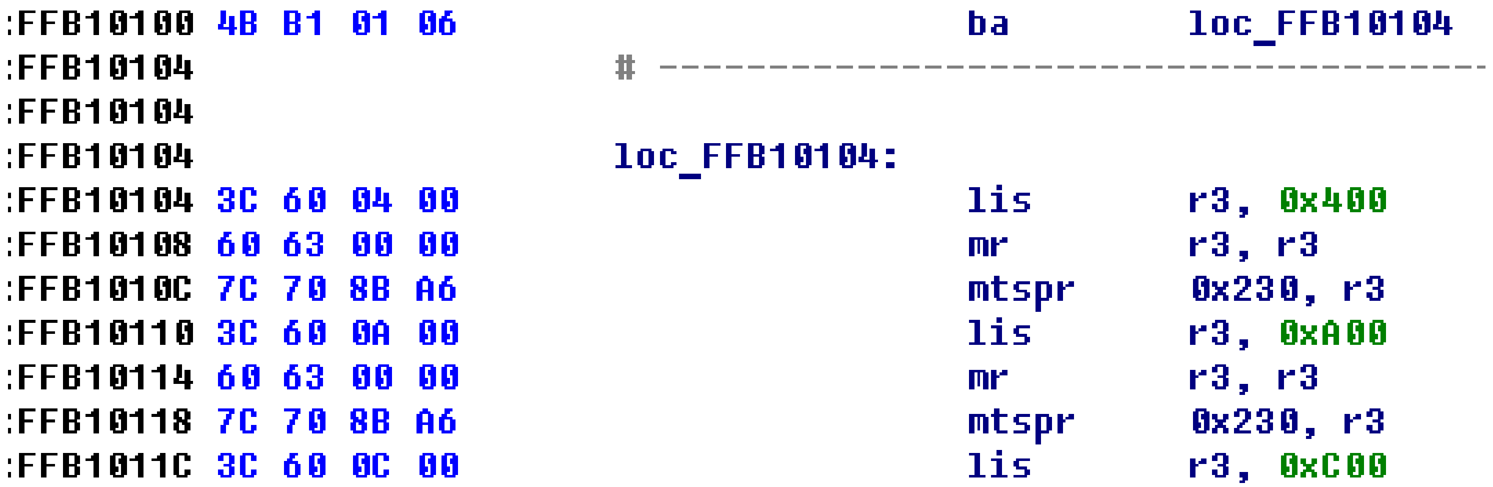
Target device - F60

The firmware is available as a file with the extension “.bin”. For example – “A09ma604.000.bin”.
At offset 0x100 starts loader code:



```
000000F0  55 55 55 55 55 55 55 55  55 55 55 55 55 55 55 55  UUUUUUUUUUUUUUUU
00000100  4B B1 01 06 3C 60 04 00  60 63 00 00 7C 70 8B A6  K±  <`  `c  |p‹¦
00000110  3C 60 0A 00 60 63 00 00  7C 70 8B A6 3C 60 0C 00  <`  `c  |p‹¦<`
```

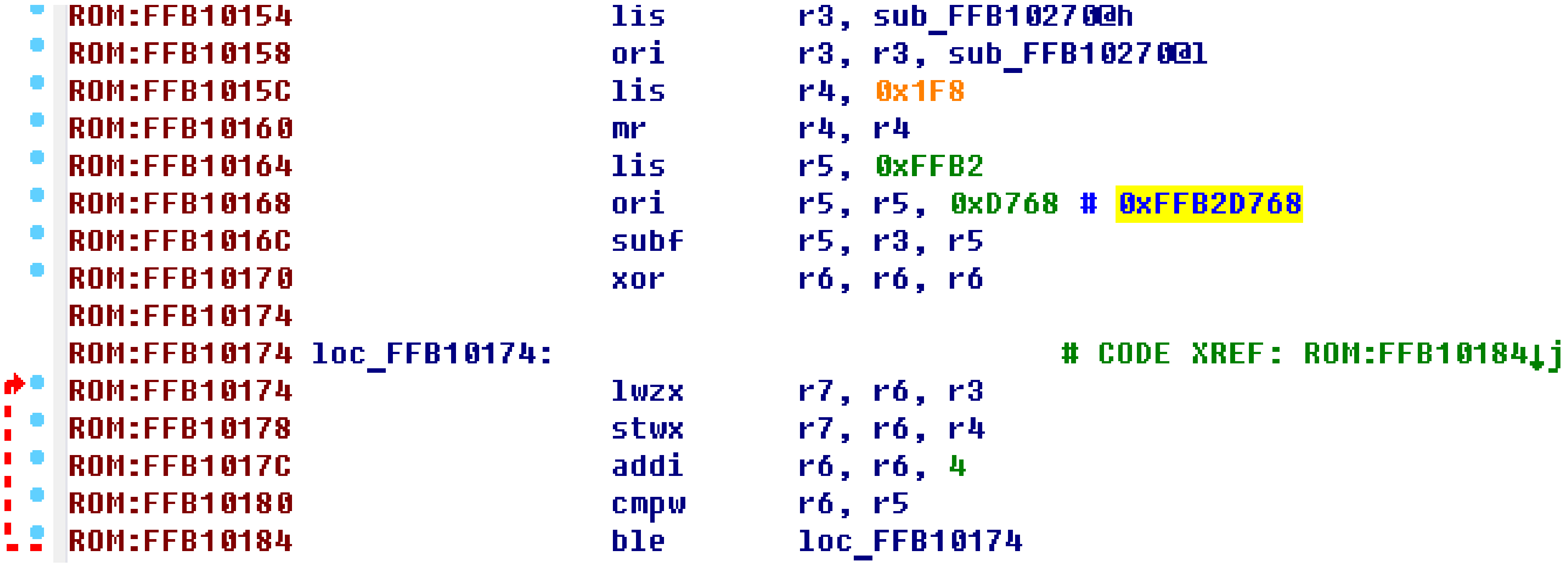
First instruction – “ba 0xFFB10104”, therefore, the base address of firmware is a 0xFFB10104.



```
:FFB10100 4B B1 01 06                    ba      loc_FFB10104
:FFB10104                 # ------------------------------------------
:FFB10104
:FFB10104                 loc_FFB10104:
:FFB10104 3C 60 04 00                    lis     r3, 0x400
:FFB10108 60 63 00 00                    mr      r3, r3
:FFB1010C 7C 70 8B A6                    mtspr   0x230, r3
:FFB10110 3C 60 0A 00                    lis     r3, 0xA00
:FFB10114 60 63 00 00                    mr      r3, r3
:FFB10118 7C 70 8B A6                    mtspr   0x230, r3
:FFB1011C 3C 60 0C 00                    lis     r3, 0xC00
```

F60 Firmware unpacking (1)

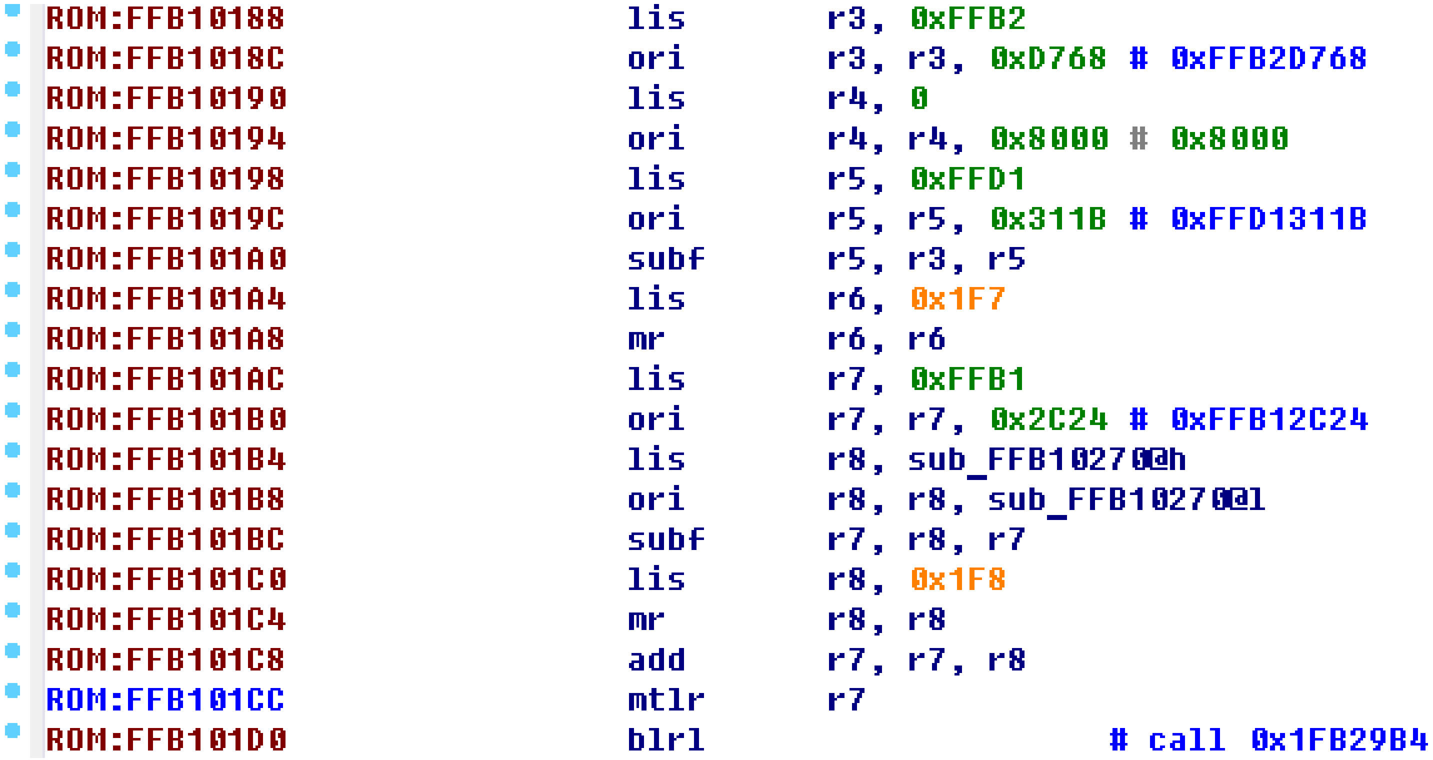
Loader copies 0x1D4F8 bytes from 0xFFB10270 to 0x1F80000.



```
ROM:FFB10154                 lis       r3, sub_FFB10270@h
ROM:FFB10158                 ori       r3, r3, sub_FFB10270@l
ROM:FFB1015C                 lis       r4, 0x1F8
ROM:FFB10160                 mr        r4, r4
ROM:FFB10164                 lis       r5, 0xFFB2
ROM:FFB10168                 ori       r5, r5, 0xD768 # 0xFFB2D768
ROM:FFB1016C                 subf      r5, r3, r5
ROM:FFB10170                 xor       r6, r6, r6
ROM:FFB10174
ROM:FFB10174 loc_FFB10174:                               # CODE XREF: ROM:FFB10184↓j
ROM:FFB10174                 lwzx      r7, r6, r3
ROM:FFB10178                 stwx      r7, r6, r4
ROM:FFB1017C                 addi      r6, r6, 4
ROM:FFB10180                 cmpw      r6, r5
ROM:FFB10184                 ble       loc_FFB10174
```

F60 Firmware unpacking (2)

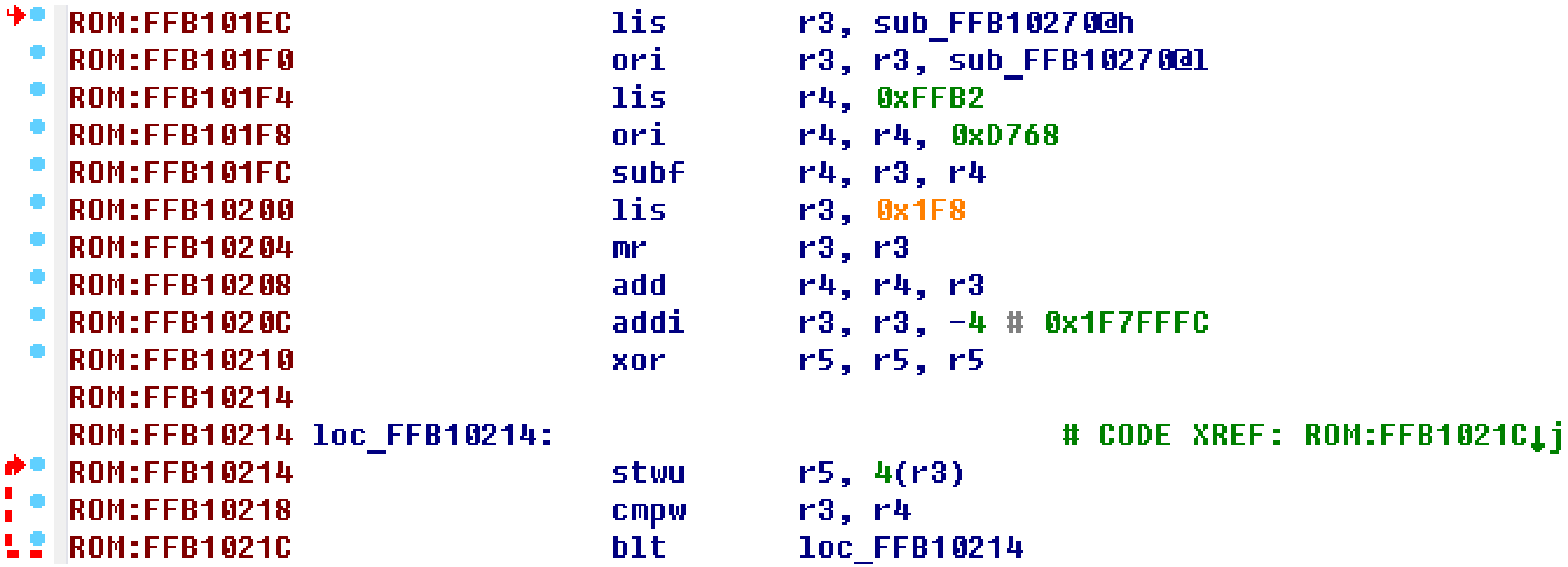
This bytes contains zlib uncompress code that unpack main firmware code from 0xFFB2D768 to 0x8000.



```
ROM:FFB10188                 lis       r3, 0xFFB2
ROM:FFB1018C                 ori       r3, r3, 0xD768 # 0xFFB2D768
ROM:FFB10190                 lis       r4, 0
ROM:FFB10194                 ori       r4, r4, 0x8000 # 0x8000
ROM:FFB10198                 lis       r5, 0xFFD1
ROM:FFB1019C                 ori       r5, r5, 0x311B # 0xFFD1311B
ROM:FFB101A0                 subf      r5, r3, r5
ROM:FFB101A4                 lis       r6, 0x1F7
ROM:FFB101A8                 mr        r6, r6
ROM:FFB101AC                 lis       r7, 0xFFB1
ROM:FFB101B0                 ori       r7, r7, 0x2C24 # 0xFFB12C24
ROM:FFB101B4                 lis       r8, sub_FFB10270@h
ROM:FFB101B8                 ori       r8, r8, sub_FFB10270@l
ROM:FFB101BC                 subf      r7, r8, r7
ROM:FFB101C0                 lis       r8, 0x1F8
ROM:FFB101C4                 mr        r8, r8
ROM:FFB101C8                 add       r7, r7, r8
ROM:FFB101CC                 mtlr      r7
ROM:FFB101D0                 blrl                    # call 0x1FB29B4
```

F60 Firmware unpacking (3)

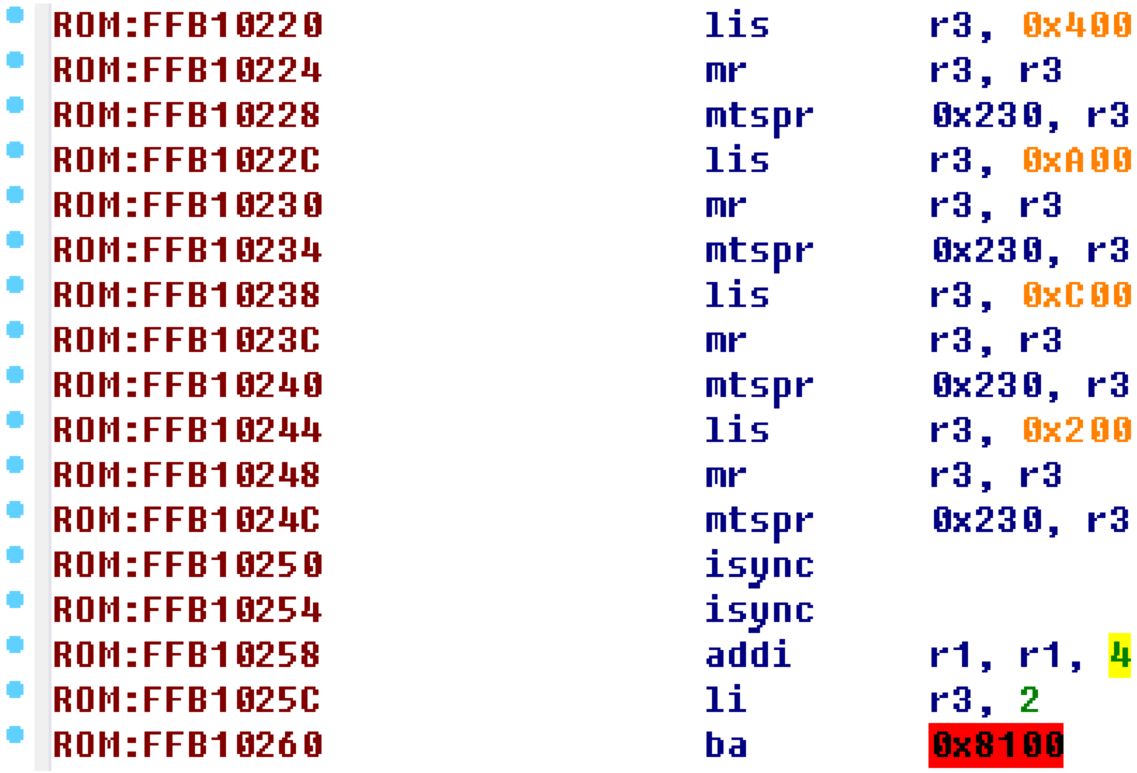
If uncompressing is finished successfully, code at offset 0x1F80000 is cleared



```
ROM:FFB101EC                  lis       r3, sub_FFB10270@h
ROM:FFB101F0                  ori       r3, r3, sub_FFB10270@l
ROM:FFB101F4                  lis       r4, 0xFFB2
ROM:FFB101F8                  ori       r4, r4, 0xD768
ROM:FFB101FC                  subf      r4, r3, r4
ROM:FFB10200                  lis       r3, 0x1F8
ROM:FFB10204                  mr        r3, r3
ROM:FFB10208                  add       r4, r4, r3
ROM:FFB1020C                  addi      r3, r3, -4 # 0x1F7FFFC
ROM:FFB10210                  xor       r5, r5, r5
ROM:FFB10214
ROM:FFB10214 loc_FFB10214:                                   # CODE XREF: ROM:FFB1021C↓j
ROM:FFB10214                  stwu      r5, 4(r3)
ROM:FFB10218                  cmpw      r3, r4
ROM:FFB1021C                  blt       loc_FFB10214
```

F60 Firmware unpacking (4)

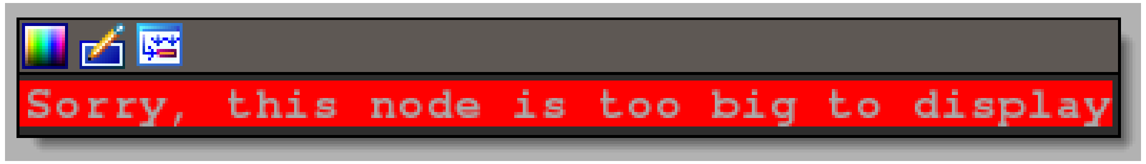
Finally, control is passed at offset 0x8100 in the uncompressed code.



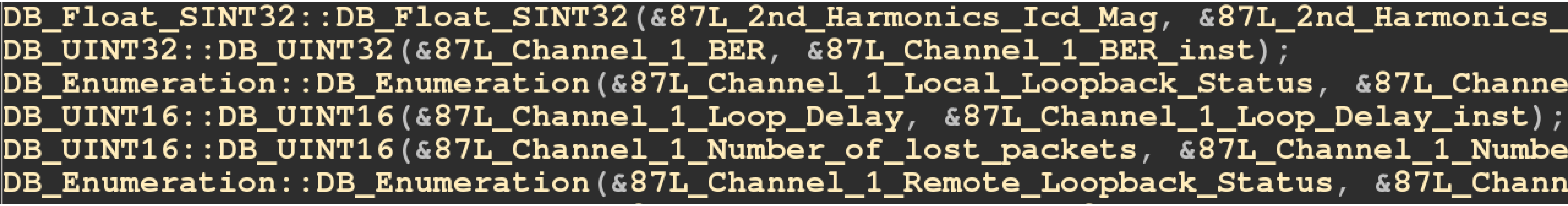
```
ROM:FFB10220                 lis      r3, 0x400
ROM:FFB10224                 mr       r3, r3
ROM:FFB10228                 mtspr    0x230, r3
ROM:FFB1022C                 lis      r3, 0xA00
ROM:FFB10230                 mr       r3, r3
ROM:FFB10234                 mtspr    0x230, r3
ROM:FFB10238                 lis      r3, 0xC00
ROM:FFB1023C                 mr       r3, r3
ROM:FFB10240                 mtspr    0x230, r3
ROM:FFB10244                 lis      r3, 0x200
ROM:FFB10248                 mr       r3, r3
ROM:FFB1024C                 mtspr    0x230, r3
ROM:FFB10250                 isync
ROM:FFB10254                 isync
ROM:FFB10258                 addi     r1, r1, 4
ROM:FFB1025C                 li       r3, 2
ROM:FFB10260                 ba       0x8100
```

Global Device Objects

- Thousands of them
- Backed up by EEPROM
- Inheritance level ~ 3
- Strongly typed => Unified access



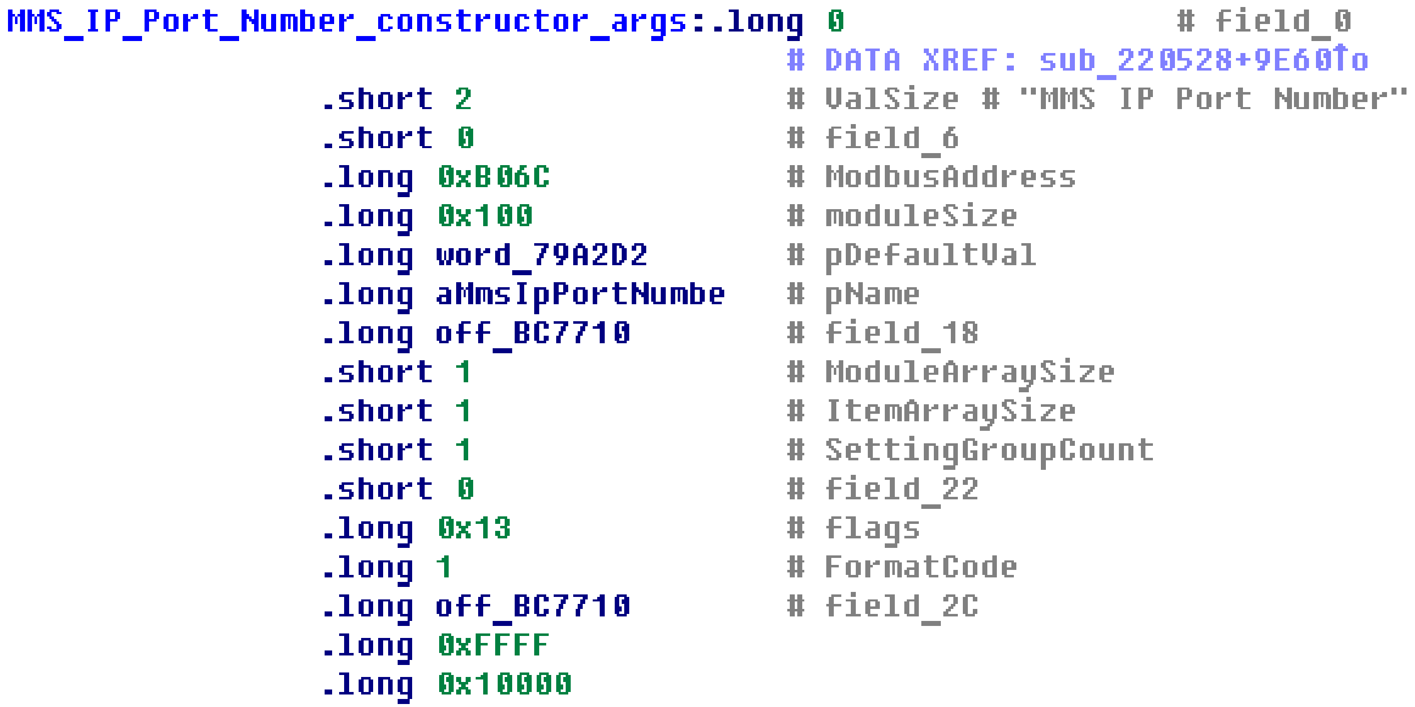
Sorry, this node is too big to display



```
DB_Float_SINT32::DB_Float_SINT32(&87L_2nd_Harmonics_Icd_Mag, &87L_2nd_Harmonics_
DB_UINT32::DB_UINT32(&87L_Channel_1_BER, &87L_Channel_1_BER_inst);
DB_Enumeration::DB_Enumeration(&87L_Channel_1_Local_Loopback_Status, &87L_Channe
DB_UINT16::DB_UINT16(&87L_Channel_1_Loop_Delay, &87L_Channel_1_Loop_Delay_inst);
DB_UINT16::DB_UINT16(&87L_Channel_1_Number_of_lost_packets, &87L_Channel_1_Numbe
DB_Enumeration::DB_Enumeration(&87L_Channel_1_Remote_Loopback_Status, &87L_Chann
```

Example of such object

This values has db based view, that initialized using hardcoded value descriptions.



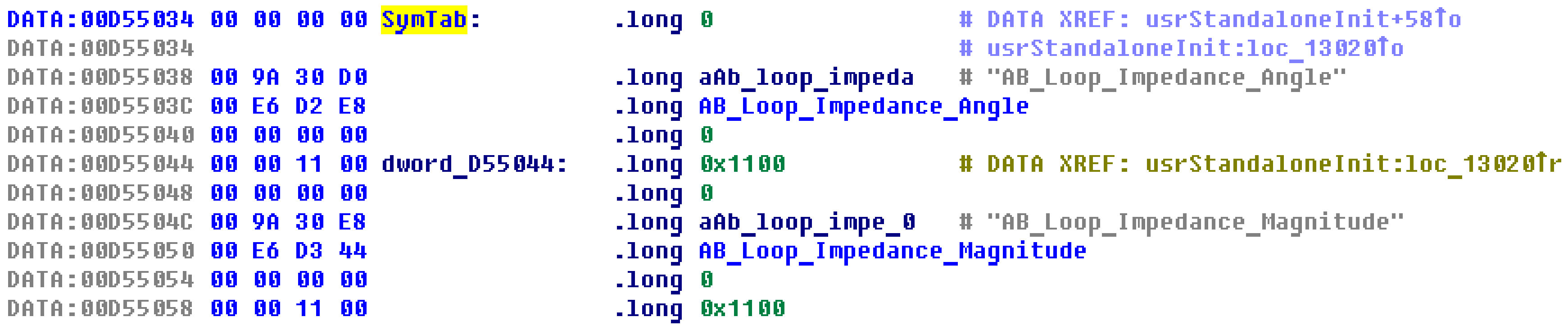
```
MMS_IP_Port_Number_constructor_args:.long 0                   # field_0
                                         # DATA XREF: sub_220528+9E60↑o
                .short 2                 # ValSize # "MMS IP Port Number"
                .short 0                 # field_6
                .long 0xB06C             # ModbusAddress
                .long 0x100              # moduleSize
                .long word_79A2D2        # pDefaultVal
                .long aMmsIpPortNumbe    # pName
                .long off_BC7710         # field_18
                .short 1                 # ModuleArraySize
                .short 1                 # ItemArraySize
                .short 1                 # SettingGroupCount
                .short 0                 # field_22
                .long 0x13               # flags
                .long 1                  # FormatCode
                .long off_BC7710         # field_2C
                .long 0xFFFF
                .long 0x10000
```

Sometimes, the new version is really better

At start we were analyzing firmware version 6.04. So, on vendor’s website has newer one.

New in firmware v. 7.31:

- VxWorks 6.8
- And that has VxWorks symbols!

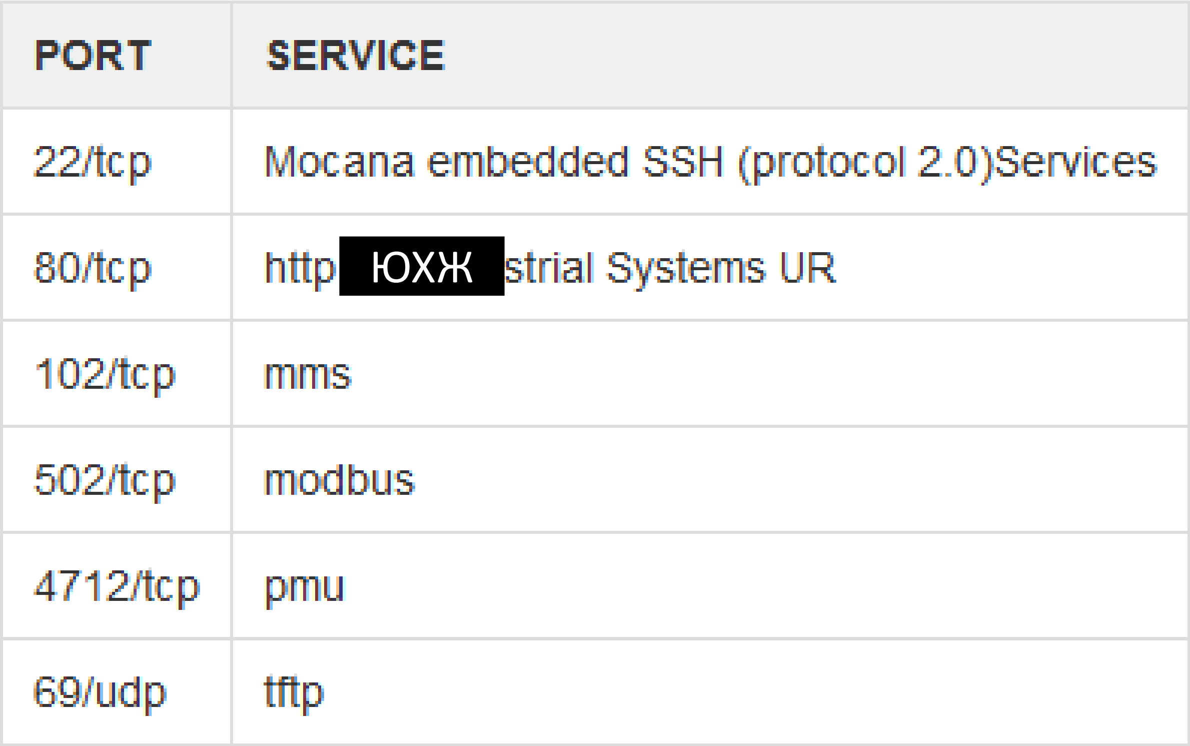


```
DATA:00D55034 00 00 00 00 SymTab:         .long 0                 # DATA XREF: usrStandaloneInit+58↑o
DATA:00D55034                                                     # usrStandaloneInit:loc_13020↑o
DATA:00D55038 00 9A 30 D0                 .long aAb_loop_impeda   # "AB_Loop_Impedance_Angle"
DATA:00D5503C 00 E6 D2 E8                 .long AB_Loop_Impedance_Angle
DATA:00D55040 00 00 00 00                 .long 0
DATA:00D55044 00 00 11 00 dword_D55044:   .long 0x1100            # DATA XREF: usrStandaloneInit:loc_13020↑r
DATA:00D55048 00 00 00 00                 .long 0
DATA:00D5504C 00 9A 30 E8                 .long aAb_loop_impe_0   # "AB_Loop_Impedance_Magnitude"
DATA:00D55050 00 E6 D3 44                 .long AB_Loop_Impedance_Magnitude
DATA:00D55054 00 00 00 00                 .long 0
DATA:00D55058 00 00 11 00                 .long 0x1100
```

Well, knowledge of the names of functions and global variables really doing life better

Services

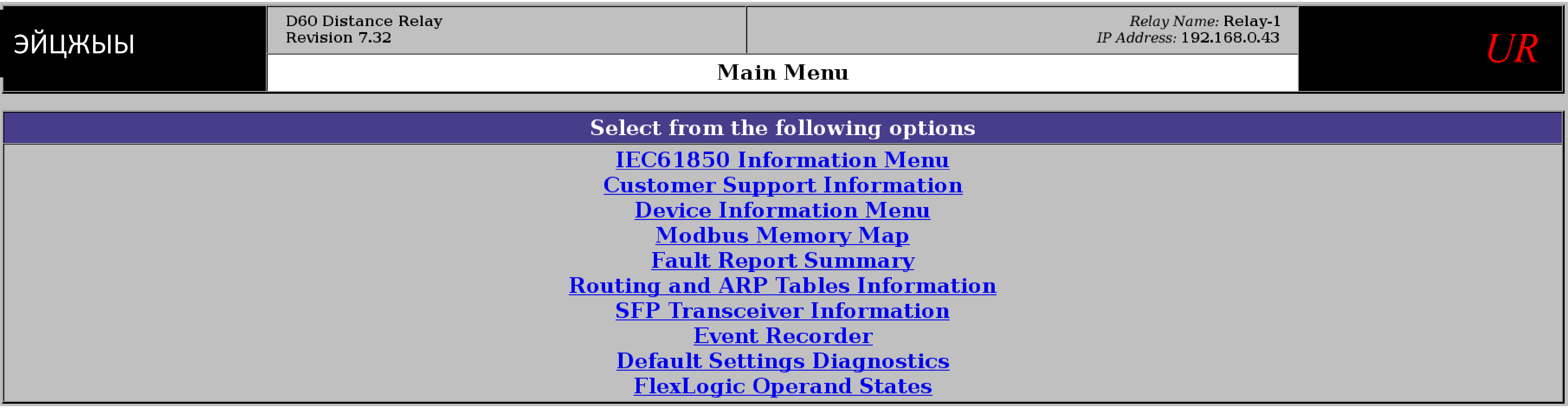
Firmware 7.31



PORT	SERVICE
22/tcp	Mocana embedded SSH (protocol 2.0)Services
80/tcp	http [redacted]strial Systems UR
102/tcp	mms
502/tcp	modbus
4712/tcp	pmu
69/udp	tftp

Simple web service

- Very simple
- No user interaction ☹



ЭЙЦЖЫЫ

D60 Distance Relay
Revision 7.32

Relay Name: Relay-1
IP Address: 192.168.0.43

UR

Main Menu

Select from the following options

[IEC61850 Information Menu]
[Customer Support Information]
[Device Information Menu]
[Modbus Memory Map]
[Fault Report Summary]
[Routing and ARP Tables Information]
[SFP Transceiver Information]
[Event Recorder]
[Default Settings Diagnostics]
[FlexLogic Operand States]

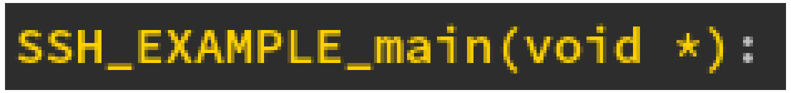
Modbus

- Authorization
  - Different modes
  - Password is a 32 bit number or username with password
  - Bruteforce protection
- R/W Access control
- Old Enervista protocol

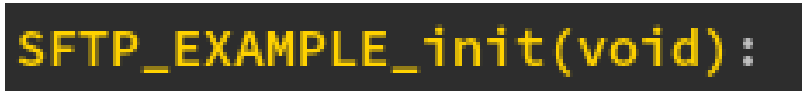
New Modbus

- New Enervista protocol
- SSH tunnel
- MocanaSSH

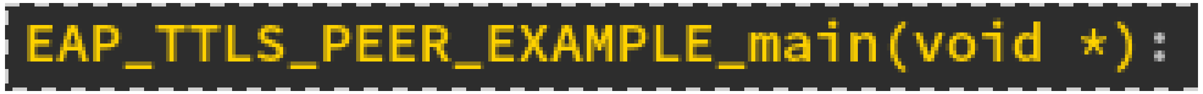
Implementing SSH



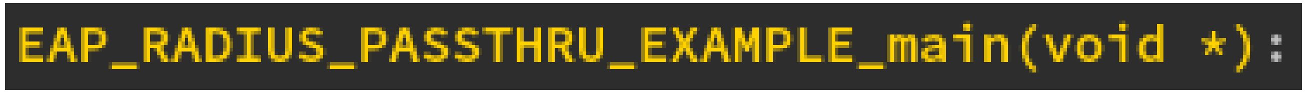
```
SSH_EXAMPLE_main(void *):
```



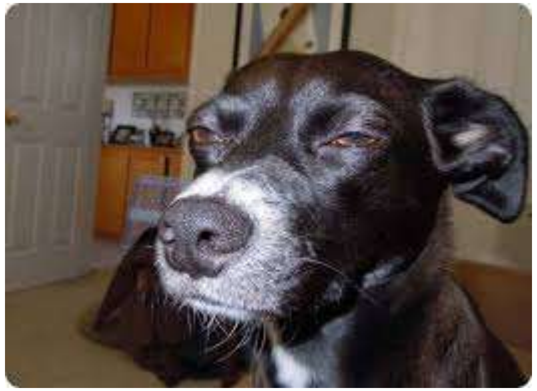
```
SFTP_EXAMPLE_init(void):
```



```
EAP_TTLS_PEER_EXAMPLE_main(void *):
```



```
EAP_RADIUS_PASSTHRU_EXAMPLE_main(void *):
```

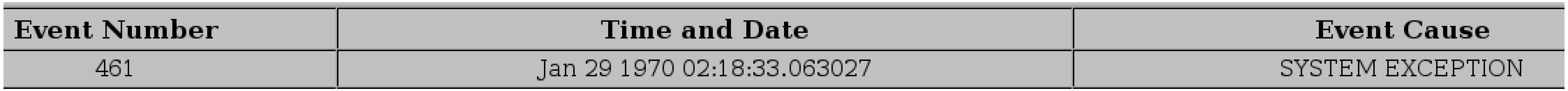


Secure CyberSecurity

- No response
  - Reported 26 Jul 2016
  - Got 4 potential RCE

No demo

- No debugger
- No crash dump
- No JTAG
- No UART
- Nothing at all



Event Number	Time and Date	Event Cause
461	Jan 29 1970 02:18:33.063027	SYSTEM EXCEPTION

REF630

- “DB based”
- FTP – full access to flash
- HTTP
- IEC 61850
- ODBC

Comfortable terminal

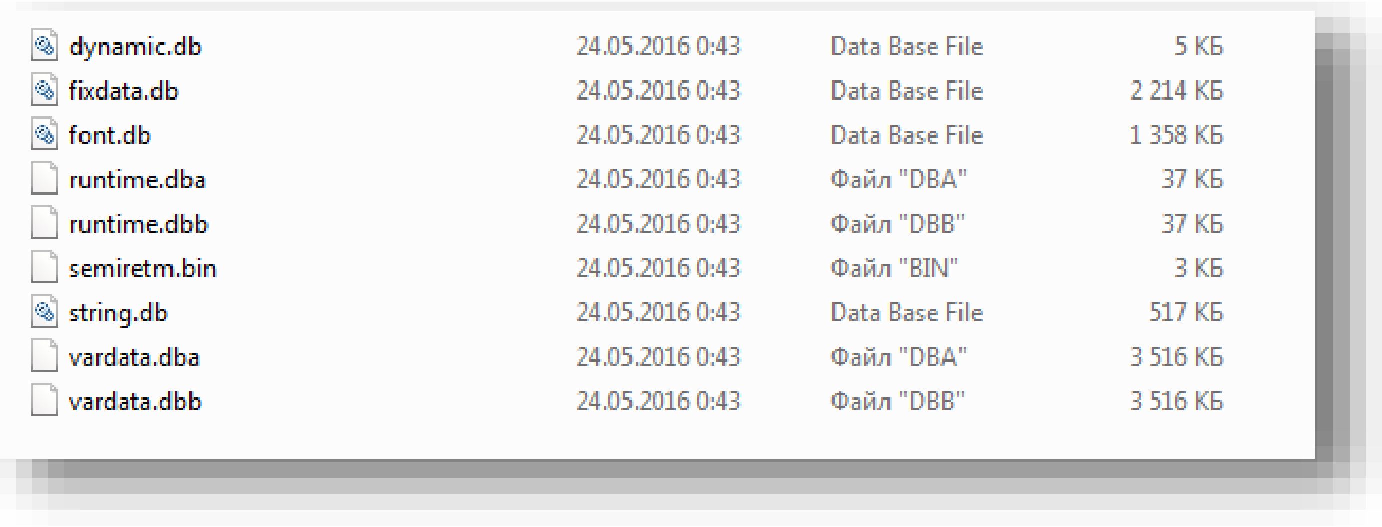
- VxWorks
- PowerPC
- FS access
- VxWorks img is ELF
- Symbols
- Traceback with PC and LR
- And something more...

Comfortable terminal

- debugsrv
  - 7755/tcp – stdout with additionals headers
  - 7766/tcp – stdin
  - Can be switched in boot
- VxWorks console
  - Internal debugger
  - Arbitrary calls by name and by address
  - Many more

Ref630 DB based

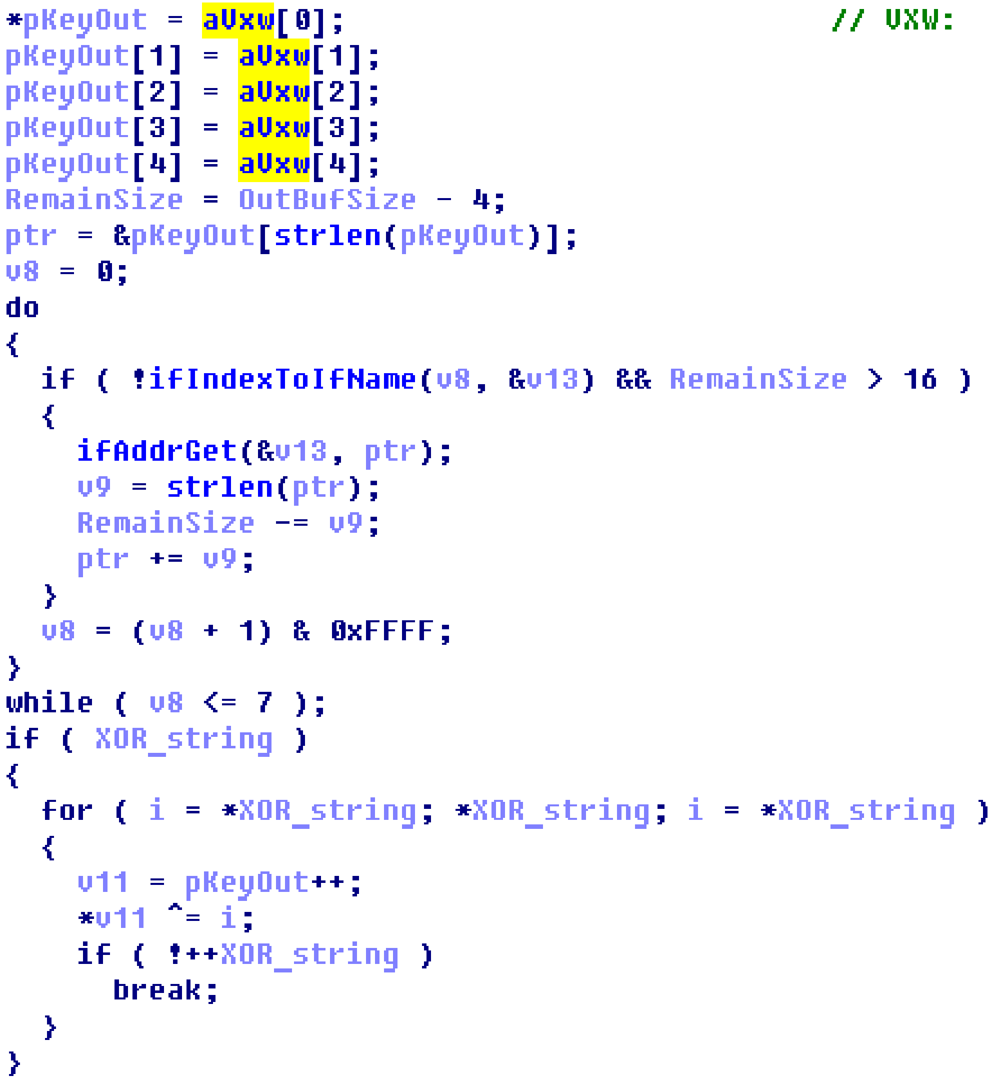
- All data in DB that is stored on file system
- Database files are divided into three types:
  - Basic – plain data, no encryption and compression
  - Sequential – compressed data blocks
  - Secure sequential – compressed and encrypted data blocks



dynamic.db	24.05.2016 0:43	Data Base File	5 КБ
fixdata.db	24.05.2016 0:43	Data Base File	2 214 КБ
font.db	24.05.2016 0:43	Data Base File	1 358 КБ
runtime.dba	24.05.2016 0:43	Файл "DBA"	37 КБ
runtime.dbb	24.05.2016 0:43	Файл "DBB"	37 КБ
semiretm.bin	24.05.2016 0:43	Файл "BIN"	3 КБ
string.db	24.05.2016 0:43	Data Base File	517 КБ
vardata.dba	24.05.2016 0:43	Файл "DBA"	3 516 КБ
vardata.dbb	24.05.2016 0:43	Файл "DBB"	3 516 КБ

Ref630 Encrypted DB files

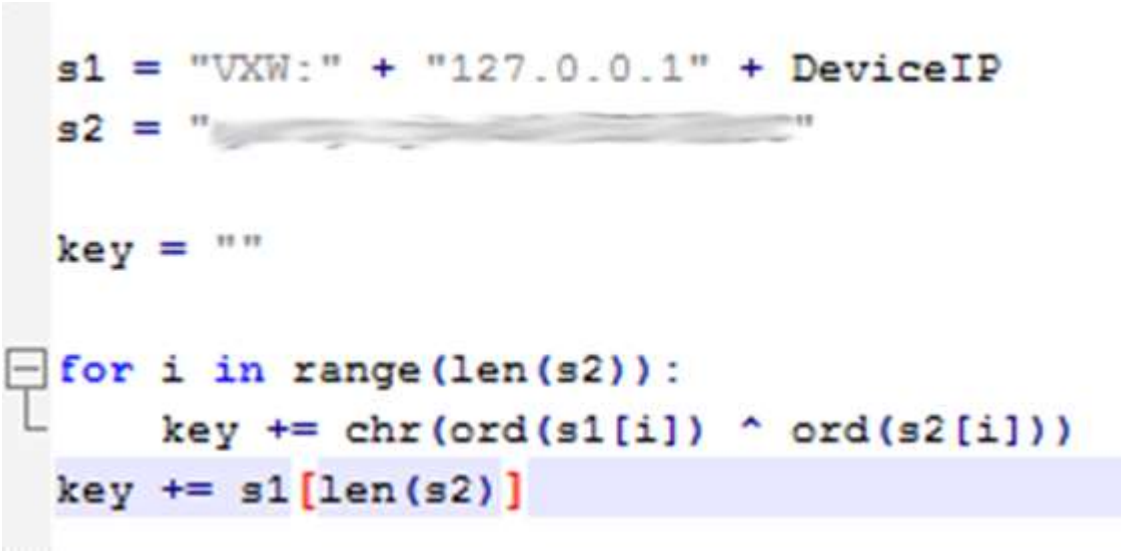
- Blowfish algorithm
- Encryption key depended on interfaces IP addresses



```
*pKeyOut = aVxw[0];                           // VXW:
pKeyOut[1] = aVxw[1];
pKeyOut[2] = aVxw[2];
pKeyOut[3] = aVxw[3];
pKeyOut[4] = aVxw[4];
RemainSize = OutBufSize - 4;
ptr = &pKeyOut[strlen(pKeyOut)];
v8 = 0;
do
{
  if ( !ifIndexToIfName(v8, &v13) && RemainSize > 16 )
  {
    ifAddrGet(&v13, ptr);
    v9 = strlen(ptr);
    RemainSize -= v9;
    ptr += v9;
  }
  v8 = (v8 + 1) & 0xFFFF;
}
while ( v8 <= 7 );
if ( XOR_string )
{
  for ( i = *XOR_string; *XOR_string; i = *XOR_string )
  {
    v11 = pKeyOut++;
    *v11 ^= i;
    if ( !++XOR_string )
      break;
  }
}
```

Ref630 Encrypted DB files

- Two interfaces
  - Loopback with IP address 127.0.0.1
  - Common with external IP address
- Hardcoded string



```
s1 = "VXW:" + "127.0.0.1" + DeviceIP
s2 = "[redacted]"

key = ""

for i in range(len(s2)):
    key += chr(ord(s1[i]) ^ ord(s2[i]))
key += s1[len(s2)]
```

Ref630 ODBC protocol

- Releases!
- Parser
  - https://github.com/rigmar/Recon2017/tree/master/DBS
- Client
  - https://github.com/rigmar/Recon2017/tree/master/ODBC

IEC 61850

- MMS Lite from SISCO
- Cares about security
- Some info about secpatches
- But “SISCO does not provide detailed technical information of any kind (security related or otherwise) on our products to anonymous or unknown persons”

MMS Lite

- No sources
- Some dumb fuzzing => No results
- Again some bug in user-hooks

Path traversal

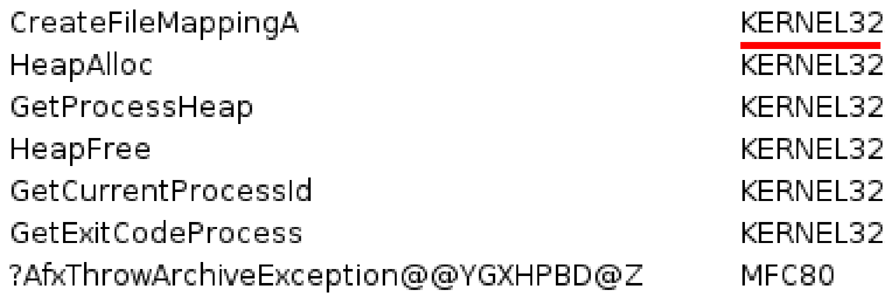
- u_mvl_fopen_ind
  - Used to read COMTRADE files
- But allows to read any file on flash
- Reported 26 Jul 2016
- Device credentials
  - stored in DB file
  - Hashed with MD5

Pimp my term!

- Schneider Electric
- Fancy TV for your terminal
- 220 Service ready on KAOS system
- Magelis xbtgt5330
- Only one default port

Firmware as OS

- Firmware consists of several .dlm files
- .dml – is ordinary PE
- x86 based
- Some kind of KAOS system
- But KAOS looks like Windows App
- PTC Perc “Real-Time” Java machine



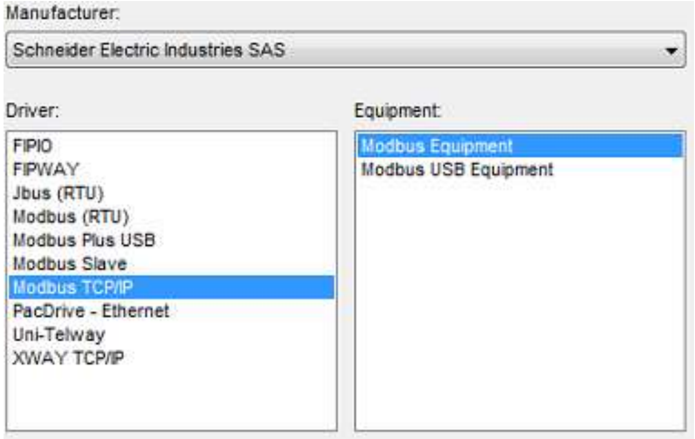
```
CreateFileMappingA                     KERNEL32
HeapAlloc                              KERNEL32
GetProcessHeap                         KERNEL32
HeapFree                               KERNEL32
GetCurrentProcessId                    KERNEL32
GetExitCodeProcess                     KERNEL32
?AfxThrowArchiveException@@YGXHPBD@Z   MFC80
```

Device management

- Vijeo management tool
- Works through FTP
- FTP has some proprietary extensions
  - TGID
  - WRDI
  - ...

Smart TV

- Can be integrated with bunch of terminals
- Some vendors even recommend it
- A lot of SW extensions



Manufacturer:

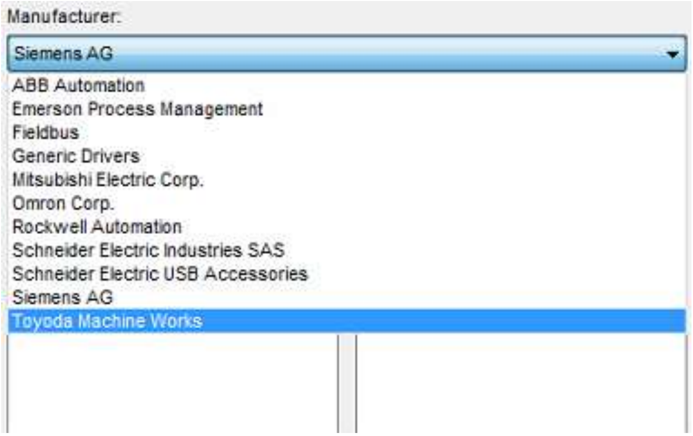
Schneider Electric Industries SAS

Driver:

FIPIO
FIPWAY
Jbus (RTU)
Modbus (RTU)
Modbus Plus USB
Modbus Slave
Modbus TCP/IP
PacDrive - Ethernet
Uni-Telway
XWAY TCP/IP

Equipment:

Modbus Equipment
Modbus USB Equipment



Manufacturer:

Siemens AG

ABB Automation
Emerson Process Management
Fieldbus
Generic Drivers
Mitsubishi Electric Corp.
Omron Corp.
Rockwell Automation
Schneider Electric Industries SAS
Schneider Electric USB Accessories
Siemens AG
Toyoda Machine Works

Augmented Smart TV

- A lot of HW extensions
- USB biometric switches

Third-party party

- Almost every IED (with IEC61850) uses SISCO MMSLite
- Mocana SSH
- Allegro ROM Pager
- Third-party soft is Good
- Update problems

Substation-ng

- Remove embedded devices
  - Goodbye, VxWorks!
  - Goodbye, PowerPC!
- Signal acquiring from power lines still required
- Put all protection processing in virtual machines
  - Application running on Windows box
- Only HI-TECH countries

In the end

- Still just an embedded device
- Real-Time requirements
  - No encryption
  - No exploit mitigations
- Updates are slow/manual/hard
- A lot of people still writing their own HTTP Servers

@scadasl kudos

@atimorin Alexander Timorin
@_Rigmar_ Alexander Tlyapov
@arbitrarycode Alexander Zaitsev
@GiftsUngiven Alexey Osipov

Anatoly Katushin
@repdet Gleb Gritsai
Sergey Gordeychik
Sergey Sidorov

iGrids Lab
Maksim Nikandrov
Viktor Nikitin
And others

http://scadastrangelove.blogspot.com

iGrids Lab

- Cheboksary, home of ‘Bouquet of Chuvashia’ beer
  - https://en.wikipedia.org/wiki/Chuvashia
- Substation (“releyka”) capital of RF
- Certification laboratory
- (ad) Access to numerous substation devices by subscription
  - (russian) http://igrids.ru/
- Open challenges on conferences

Thanks for Your attention